

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль

(підпис)

(ініціали, прізвище)

“ _____ ” _____ 2018р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки та інформаційні технології
за спеціалізацією - Геометричне моделювання в інформаційних системах
на тему: Моделювання процесів деформації геометричних об'єктів

Виконав (-ла): студент (-ка) 6 курсу, групи ТР-71м

Шпикуляк Олександр Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник к.т.н., доц. Сидеренко. Ю.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

(підпис)

Київ - 2018

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки та інформаційні технології

за спеціалізацією - Геометричне моделювання в інформаційних системах

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали) (підпис)
«____» _____ 2018р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Шпикуляк Олександр Олександрович

(прізвище, ім'я, по батькові)

1. Тема дисертації Моделювання процесів деформації геометричних об'єктів

Науковий керівник Сидоренко Ю.В., к.т.н., доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “__” _____ 20__ року №__

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження: є комп'ютерні технології геометричного моделювання

4. Предмет дослідження: є комп'ютерні технології геометричного моделювання з використанням політочкових перетворень

5. Перелік питань, які потрібно розробити:

– проаналізувати існуючі методи геометричного моделювання об'єктів;

– дослідити особливості та визначити властивості політочкових перетворень;

– порівняти результати проведення політочкових перетворень певної області з областю, заданою за допомогою функції Гауса

6. Орієнтований перелік ілюстративного матеріалу: презентація на тему «Система моделювання процесів деформації геометричних об'єктів»

7. Орієнтований перелік публікацій: Сидоренко Ю.В. Система моделювання геометричних об'єктів за допомогою політочкових перетворень [Текст] / Сидоренко Ю.В. // Міжвідомчий науково-технічний збірник „Прикладна геометрія та інженерна графіка”.-Вип.92.-К.:КНУБА, 2016р.- С.118-125.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада	Підпис, дата
--------	------------------------------	--------------

	консультанта	завдання видав	завдання прийняв
Основна частина	Сидоренко Ю.В., доц., к.т.н.		

9. Дата видачі завдання «_____» _____ 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	19.01.18 р.	
2	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	20.01.18р. — 28.01.18р.	
3	Підготовка матеріалів магістерської роботи	12.10.18р. — 21.11.18р.	
4	Підготовка публікацій	03.11.18р. — 12.11.18р.	
5	Доповідь на конференції	13.11.18р.	
6	Написання основних розділів автореферату	6.11.18р. — 20.11.18р.	
7	Захист програмного продукту	24.10.18р	
8	Передзахист	31.11.18р	
9	Захист	17.12.18р	

Студент

(підпис)

Шпикуляк О.О.

(прізвище та ініціали)

Науковий керівник

(підпис)

Сидоренко Ю.В.

(прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація складається зі вступу, шести розділів, висновку, списку джерел, містить 25 рисунків. Повний обсяг магістерської дисертації складає 116 сторінки.

Моделювання деформацій є одним з напрямків роботи з динамічними об'єктами, в якому під впливом визначених процесів відбуваються динамічні зміни. Під час моделювання таких процесів виникає проблема конструювання об'єктів, що зазнають деформаційних змін. В сучасній практиці моделювання геометричних динамічних об'єктів вимагається застосування таких методів, які дозволяли б провести деформацію без безпосередньої взаємодії з об'єктом, визначаючи лише параметри динамічної деформації і не залежали від певного виду функціонального визначення геометричного об'єкта. Цю задачу можуть вирішити політочкові перетворення, дозволяючи інтерактивно деформувати геометричний об'єкт.

Пояснювальна записка складається із вступу, 6 розділів, висновку, переліку посилань та додатків.

У вступі викладена актуальність магістерської дисертації. Перший розділ містить відомості про мету роботи, об'єкт та предмет досліджень, вхідні та вихідні дані. Другий розділ містить аналіз існуючих програмних продуктів, які використовуються для деформації геометричних об'єктів. У третьому розділі викладено математичний апарат полікоординатних перетворень. У четвертому розділі описано програмний інструментарій розробки системи. П'ятий розділ містить опис програмної реалізації, функціональну схему системи, та містить сценарії роботи користувача з програмою. Шостий розділ присвячений розробці стартапу.

Розроблена програма дозволяє візуально представляти результати роботи. Математичний апарат описано докладно і зрозуміло. Матеріал у пояснювальній записці викладено у логічній послідовності.

Магістерська робота виконана на високому рівні, відповідає поставленій задачі та вимогам до дипломної роботи магістра. Програмна реалізація запропонованих у роботі методів та алгоритмів виконана у повному обсязі. Тому, на мою думку, магістерська робота заслуговує оцінку «відмінно», а Шпикуляк О. О. заслуговує на присвоєння кваліфікації аналітик комп'ютерних систем за спеціальністю 122 - Комп'ютерні науки та інформаційні технології.

Ключові слова. деформації геометричних об'єктів, апарат полікоординатних перетворень, програмний інструментарій, сценарії роботи

ЗМІСТ

Вступ.....	11
1. Задача моделювання процесів деформації геометричних об'єктів	13
2. Аналіз існуючих способів моделювання процесів деформації геометричних об'єктів	16
2.1. Методи інтерполяції.....	16
2.2. Методи деформаційного моделювання	24
3. Математичний апарат системи моделювання процесів деформації геометричних об'єктів	30
4. Програмний інструментарій розробки системи моделювання процесів деформації геометричних об'єктів за допомогою полікоординатних преретворень.....	48
4.1. Технології для розробки інтерфейсу	48
4.2. Інструменти для роботи з графікою. WebGL і Three.js	52
4.3. Мова програмування PHP. Фреймворк Laravel	55
4.4. Середовище Sublime Text.....	60
4.5. Модуль phpMyAdmin	63
5. Опис програмної реалізації	69
5.1. Розрахунковий модуль	70
5.2. Інтерфейсний модуль	71
5.3. Модуль візуалізації.....	72
5.4. Модульна система	72
5.5. Робота користувача з програмою	78
6. Розроблення стартап-проекту	87

	10
6.1. Опис ідеї стартап-проекту.....	87
6.2. Технологічний аудит ідеї проекту	88
6.3. Аналіз ринкових можливостей запуску стартап-проекту	89
Висновки.....	96
Список використаних джерел.....	98
Додаток А	99

ВСТУП

Техніки деформування використовуються для моделювання вигляду геометричних фігур. Однією із найважливіших проблем геометричного моделювання є реалізація операцій для зміни форми об'єкта. Деформації є потужними операціями скульптурування, оскільки вони дозволяють модифікувати форму високого рівня, на відміну від маніпулювання геометричними сутностями нижнього рівня. Особливо привабливим типом деформації є просторова деформація, яка діє на всьому просторі незалежно від представлення деформованих об'єктів всередині простору. [1]

Існує багато методів деформації геометричних об'єктів, які дають можливість змінювати об'єкти глобально або локально. Дані методи можуть буди поділені на 2 категорії : деформація що заснована на фізичній дії та геометрична деформація. Деформації, що пов'язана із фізичною дією використовуються для роботи із об'єктами згідно до натуральних законів або при апроксимації поверхні об'ємного тіла. Геометрична деформація безпосередньо деформує модель об'єкта в один із наступних способів:

а) За допомогою модифікації параметрів, які визначають модель об'єкту. Наприклад об'єкт що визначений як сплайн патчів може бути деформовано за допомогою руху його контрольних точок.

б) За допомогою застосування математичної трансформації. В цьому випадку простір у якому знаходить тіло деформується, і як наслідок спричиняє деформацію тіла. При цьому типі деформації контрольні точки є точками поверхні або простору у якому знаходиться об'єкт, а не точками об'єкту.

Геометричні деформації часто імплементуються у інтерактивному програмного забезпеченні для моделювання. Користувач деформує початковий

простий об'єкт у більш складний і при цьому він повністю контролює деформацію у режимі реального часу. [12]

У зв'язку з цим виникає низка задач у галузі прикладної геометрії, розв'язання яких дозволить ефективно використовувати математичну базу для розробки та впровадження високоефективних автоматизованих систем.

Області застосування результатів вирішення задач моделювання деформації об'єктів варіюються і можуть бути використані у найрізноманітніших діяльностях – від моделювання поширення лісових пожеж до моделювання дизайнів комп'ютерної гри.

Питанням що ставиться до даної дослідницької роботи – яку поведінку об'єкта будуть обумовлювати деформації різного типу, а також проаналізувати порівняти результати проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса. Для порівняння буде використовуватись відношення площ областей створеними різними методами, що може бути використано для визначення других характеристик

1. ЗАДАЧА МОДЕЛЮВАННЯ ПРОЦЕСІВ ДЕФОРМАЦІЇ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ

Геометричні перетворення є одним з найбільш розповсюджених математичних апаратів конструювання геометричних об'єктів через деформацію. Типовими задачами, що останнім часом все частіше постають при геометричному моделюванні є оперування геометричними об'єктами, які не мають точного математичного описання, керований синтез геометричних об'єктів необхідної форми за їх визначеними прототипами, отримання в процесі моделювання проміжних форм за відомими прообразом та образом перетворень, визначення закону перетворення, тощо. Ці та інші питання дозволяє вирішити запропонований в роботі спосіб деформативного конструювання, що ґрунтується на геометричному перетворенні простору, разом з яким перетворюється і геометричний об'єкт, що в ньому міститься. [1]

Однією з обов'язкових умов наукової роботи в наш час є не тільки проведення фундаментальних та прикладних досліджень на високому рівні, але й доведення наукових розробок до стадії практичного їх використання. Автоматизовані системи суттєво збільшують продуктивність праці та змінюють її характер. Базою будь-якої системи деформативного моделювання повинен бути пакет програм опису та конструювання зовнішніх геометричних форм об'єкта, що зазнав деформаційних змін. Це стосується як галузі моделювання технічних об'єктів, так і, наприклад, дослідження та обробки факторів, що пов'язані з навколишнім середовищем (інформація про розповсюдження екологічного забруднення), де необхідно моделювати стан середовища у поточний період часу (наприклад, крайки вигорання під час пожежі), проводити оцінку та отримувати показники будь-якої точки місцевості. [63] При цьому дедалі важливішою стає задача швидкого реагування на ситуацію, що склалась, коли необхідно за досить короткий час змоделювати за

допомогою простих і наочних методів існуюче положення (наприклад, після аварії або пожежі) з точки зору надання першої допомоги і спрогнозувати наслідки того, що сталося.

Таким чином, при геометричному моделюванні об'єктів, що зазнали деформаційних змін, необхідно створити деякий керуючий апарат на базі геометричних перетворень з можливістю зорового відстеження у реальному часі результатів [64] своєї роботи, який дозволяє змінювати конфігурацію об'єкту в результаті певної сукупності дій, які є нескладними та інтуїтивно зрозумілими користувачеві.

Виходячи з вищевикладеного, створення способу політочкових перетворень для цілей деформативного моделювання, який вирішує вищезазначені питання, є актуальним. [65]

Метою магістерської роботи є дослідити математичний апарат політочкових перетворень, а саме дослідити його ефективність. Робота несе когнітивний характер. Вона ставить перед собою мету проаналізувати різні властивості політочкових перетворень та порівняти результати проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса. [64] Оскільки використовується математична модель, то немає складнощів порахувати площі цих фігур, що може бути використано для визначення інших характеристик.

Аналіз систем моделювання та деформації геометричних об'єктів показав, що існує необхідність в реалізації системи що дасть змогу досліджувати метод політочкових перетворень, що дає змогу побудувати систему з можливістю зорового відстеження результатів проведення політочкових перетворень та порівняти результати проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса. Таким чином, актуальною є розробка та вдосконалення комп'ютерних технологій геометричного моделювання.

На основі аналізу предметної області та визначення мети роботи, що полягає створенні системи моделювання деформації геометричних об'єктів, виявлення нових підходів до моделювання деформацій за допомогою удосконалення методів геометричного моделювання, було сформульовано наступні завдання досліджень:

- проаналізувати існуючі методи геометричного моделювання об'єктів;
- дослідити особливості та визначити властивості полікоординатних перетворень;
- порівняти результати проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса;

2. АНАЛІЗ ІСНУЮЧИХ СПОСОБІВ МОДЕЛЮВАННЯ ПРОЦЕСІВ ДЕФОРМАЦІЇ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ

Наразі не існує програмного продукту, що досліджує ефективність математичного апарату політочкових перетворень, аналізує різні властивості політочкових перетворень та порівнює область задану за допомогою функції Гауса з результатами проведення політочкових перетворень певної області. Робота несе когнітивний характер.

Оскільки дана робота досліджує політочкові перетворення, ми можемо розглянути методи деформативного моделювання та методи інтерполяції, як альтернативні способи моделювання процесів деформації геометричних об'єктів.

2.1. Методи інтерполяції

До стандартних методів інтерполяції можемо віднести кусково-лінійну інтерполяцію, поліноми Ньютона та Лагранжа. Використання цих методів може призвести до великої похибки підрахунків, при умові, що площа території досить велика, також за допомогою них не можливо провести інтерполяцію замкненого каркасу точок. [3]

Поліноміальні методи, як метод Лагранжа, мають деякі мінуси, які не дозволяють користуватись ним для вирішення даної задачі, зокрема погіршення результату при великій кількості точок та неоднорідності точкового базису.

Ніякий з цих методів не дає можливості інтерполювати замкнений каркас точок, що є необхідною умовою дипломної роботи. [66] Розглянемо класичні методи інтерполяції детальніше.

Кусково-лінійний метод. Нехай функція $f(x)$ задана у вигляді таблиці. Змінимо функцію на ламану лінію, у якій кожний відрізок буде сполучати усі сусідні точки таблиці.

Тоді для будь-якої точки x значення функції $y(x)$ може бути підраховано за двома сусідніми точками за наступною формулою:

$$y = y_{i-1} + (x - x_{i-1}) \frac{(y_i - y_{i-1})}{(x_i - x_{i-1})}$$

Це є формула кусково-лінійного методу інтерполяції. [5]

Таким чином, при використанні кусково-лінійного інтерполяційного методу, в першу чергу необхідно визначити інтервал, в який потрапляє значення аргументу, після цього підставити значення у формулу з коефіцієнтами, відповідними для визначеного інтервалу, і підрахувати наближене значення функції. [67]

Метод Лагранжа. Більш всього загальним виразом параболічної інтерполяції є інтерполяційна формула Лагранжа. Розглянемо многочлен $P_i(x)$, наведений у наступній формулі:

$$P_i(x_i) = 1, P_i(x_j) = 0, i \neq j, (i, j = 1, 2, \dots, n)$$

Якщо P_i буде давати 0 в точках x_1, x_2, \dots, x_n , окрім x_i , то можна його представити у наступному вигляді:

$$P_i(x) = C_i \cdot (x-x_1)(x-x_2) \cdot \dots \cdot (x-x_{i-1})(x-x_{i+1}) \cdot \dots \cdot (x-x_n),$$

де C_i – це постійний коефіцієнт.

Визначивши, що $x = x_i$, і рахуючи, що при цьому $P_i(x_i) = 1$, ми отримаємо

$$1 = C_i \cdot (x_i-x_1)(x_i-x_2) \cdot \dots \cdot (x_i-x_{i-1})(x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_n),$$

звідки можемо вивести наступне твердження:

$$C_i = \frac{1}{(x_i - x_1)(x_i - x_2) \cdot \dots \cdot (x_i - x_{i-1})(x_i - x_{i+1}) \cdot \dots \cdot (x_i - x_n)}$$

Після підстановки C_i в дану формулу ми можемо визначити кінцеву формулу підрахунку P_i (2.1):

$$P_i(x) = \frac{(x-x_1)(x-x_2) \cdot \dots \cdot (x-x_{i-1})(x-x_{i+1}) \cdot \dots \cdot (x-x_n)}{(x_i-x_1)(x_i-x_2) \cdot \dots \cdot (x_i-x_{i-1})(x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_n)} \quad (2.1)$$

Тепер перейдемо до визначення значення $P_n(x)$, що буде задовольняти умови інтерполяції, тобто приймаючого у точках y_i . Такий многочлен має наступний вигляд (2.2):

$$P_n(x) = \sum_{i=0}^n P_i(x)y_i \quad (2.2)$$

Отже, об'єднавши формули (2.1) та (2.2) отримаємо інтерполяційну формулу Лагранжа (2.3):

$$P_n(x) = \sum_{i=1}^n \frac{(x-x_1)(x-x_2) \cdot \dots \cdot (x-x_{i-1})(x-x_{i+1}) \cdot \dots \cdot (x-x_n)}{(x_i-x_1)(x_i-x_2) \cdot \dots \cdot (x_i-x_{i-1})(x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_n)} \quad (2.3)$$

При значній кількості точок даний метод має помітну похибку або при різкій зміні кривизни функції. [6]

Метод Ньютона. Нехай у однаково віддалених точках $x_i = x_0 + i \cdot h$ ($i = 0, 1, 2, \dots, n$) де h – це крок інтерполяції, задані значення $y_i = f(x_i)$ для $y = f(x)$. Потрібно визначити поліном $P_n(x)$, у якого степінь не вище n , що у буде задовольняти умову $P_n(x_i) = y_i$. [4]

Інтерполяційні поліноми зручно використовувати, коли вони задані в формі Ньютона, при умові, що точка інтерполяції буде знаходитись близько до початку

(пряма формула Ньютона) або до кінця таблиці (зворотна формула Ньютона) значень. [7]

Пряма інтерполяційна формула Ньютона представлена у вигляді:

$$P_n(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \dots + \frac{q(q-1)\dots(q-n+1)}{n!}\Delta^n y_0$$

Обернена інтерполяційна формула Ньютона має вигляд:

$$P_n(x) = y_n + q\Delta y_{n-1} + \frac{q(q+1)}{2!}\Delta^2 y_{n-2} + \dots + \frac{q(q+1)\dots(q+n-1)}{n!}\Delta^n y_0,$$

де $q = \frac{x-x_0}{h}$, вирази виду $\Delta^k y_i$ підраховується за наступною формулою:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$$

Пряму інтерполяційну формулу Ньютона потрібно використовувати при інтерполяції функції $y = f(x)$ в окрузі початкової точки x_0 , де q досить мале. Якщо дана умова не виконана, то краще використовувати обернену формулу. [68]

Метод Гауса. Нехай функція $f(x)$ буде задана дискретним каркасом точок: $y_i = f(x_i)$, тобто будуть задані пари (x_i, y_i) . Задачею інтерполяції буде побудова такої функції $\varphi(x)$, яка повинна приймати в даних точках значення y_i такі, як і функція $f(x)$, а відхилення $\varphi(x)$ від $f(x)$ в проміжних точках повинно бути максимаально малим. [9]

Функцію апроксимації побудуємо у вигляді узагальненого многочлена за наступною формулою:

$$\varphi(x) = a_{00} \cdot \psi_1(x) + a_{11} \cdot \psi_2(x) + \dots + a_{nn} \cdot \psi_n(x),$$

де $\psi_i(x)$ – система певних незалежних функцій.

В якості $\psi_i(x)$ була вибрана система експоненційних функцій. Запишемо інтерполяційну функцію, що далі ми будемо називати Гаус-функцією, що мають вигляд суми опорних функцій:

$$\varphi(x) = \sum_{i=1}^n \psi_i(x),$$

де $\psi_i(x) = \tilde{y}_i e^{-\alpha(x-x_i)^2}$, $i=1, 2, \dots, n$ — базисні значення для функції $\varphi(x)$ при заданих аргументах x_1, x_2, \dots, x_n , α — деякий коефіцієнт, значенням якого можна варіювати. [10] В даній роботі для розрахунку цього коефіцієнту було вибрано наступну формулу:

$$\alpha = \frac{\pi(n-1)}{k},$$

s

у якій $k = (x_{\max} - x_{\min})^2$, x_{\max}, x_{\min} — максимальне і мінімальне значення аргументу x , тобто значення кожного з кінців відрізка. [11]

Геометричне тлумачення інтерполяційного методу функції Гауса показана на рисунку 2.1.

Зміст методу Гаус-функції полягає в наступному.

Нехай задано дискретний каркас точок, тобто, пари вигляду (x_i, y_i) , $i = 1, 2, \dots, n$. Для даних точок необхідно знайти базисні значення $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n$, на яких будуть будуватимуться опорні функції $\psi_i(x)$ за наступною формулою:

$$\psi_i(x) = \tilde{y}_i e^{-\pi(n-1) \frac{(x-x_i)^2}{(x_{\max}-x_{\min})^2}} = \tilde{y}_i e^{-\alpha(x-x_i)^2}$$

Базисні значення можна знайти, розв'язуючи наступну систему виразів вигляду:

$$\begin{cases} \tilde{y}_1 e^{-\alpha(t_1-t_1)^2} + \tilde{y}_2 e^{-\alpha(t_1-t_2)^2} + \dots + \tilde{y}_n e^{-\alpha(t_1-t_n)^2} = y_1, \\ \tilde{y}_1 e^{-\alpha(t_2-t_1)^2} + \tilde{y}_2 e^{-\alpha(t_2-t_2)^2} + \dots + \tilde{y}_n e^{-\alpha(t_2-t_n)^2} = y_2, \\ \dots \\ \tilde{y}_1 e^{-\alpha(t_n-t_1)^2} + \tilde{y}_2 e^{-\alpha(t_n-t_2)^2} + \dots + \tilde{y}_n e^{-\alpha(t_n-t_n)^2} = y_n \end{cases}$$

Ця лінійна система, у якої матриця системи є симетричної відносно головної діагоналі. В даній матриці на головній діагоналі будуть розташовуватись одиниці, а інші елементи не будуть по модулю перевищувати одиниці. [8]

Вказана система може бути легко розв'язується будь-яким чисельним методом, в даній роботі ми розглянемо розв'язок за допомогою метода Гауса.

Отриманий вектор в майбутньому використовуватись для побудови інтерполяційної Гаус-функції.

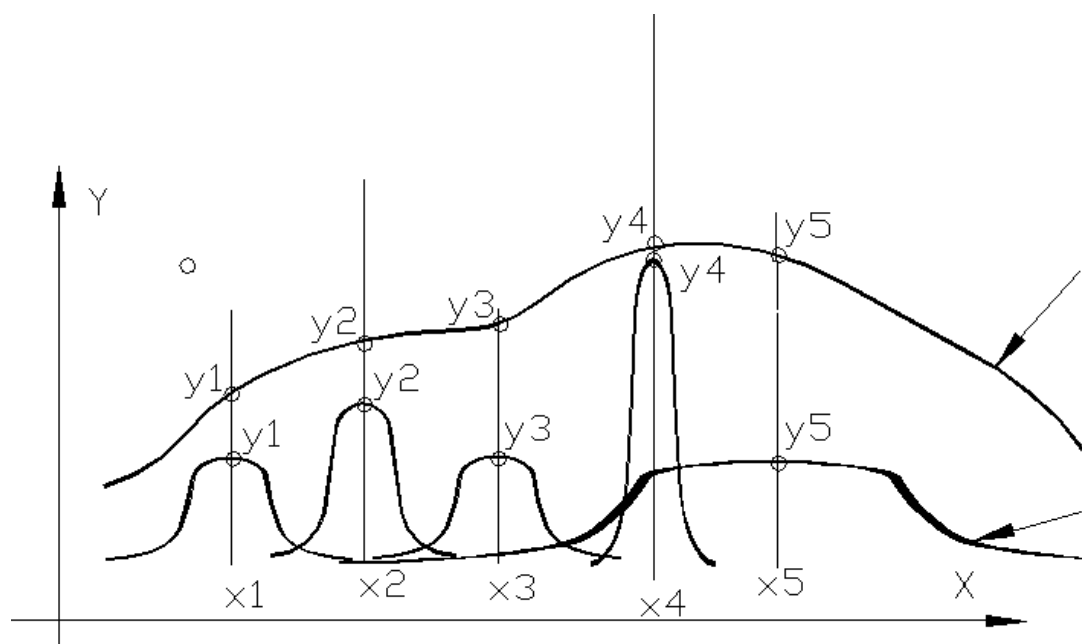


Рисунок 2.1. — Інтерполяційна функція Гауса

Вектор-розв'язок даної системи $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n$ буде використовуватись для побудови інтерполяційної Гаус-функції:

$$G(x) = \tilde{y}_1 e^{-\alpha(x-x_1)^2} + \tilde{y}_2 e^{-\alpha(x-x_2)^2} + \dots + \tilde{y}_n e^{-\alpha(x-x_n)^2}$$

Даний спосіб інтерполяції відрізняється від інших перш за все тим, що вплив будь-якого відхилення в експоненціальній залежності буде зменшуватись з зростанням відстані, що буде зумовлювати краще наближення в порівнянні з методом Лагранжа (рисунок 2.2). [13]

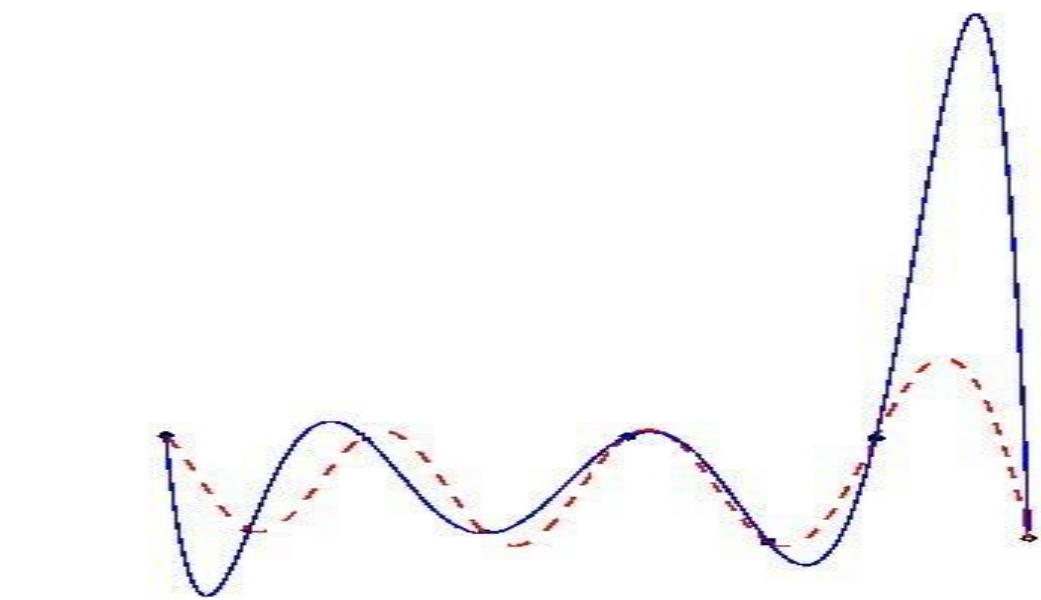


Рисунок 2.2. — Порівняльний аналіз інтерполяційної функції Гауса та полінома Лагранжа:

- пунктирна лінія – інтерполяційна функція Гауса;
- суцільна лінія – інтерполяційний поліном Лагранжа.

Також, інтерполяційна функція Гауса є n -раз диференційованою і є стійкою до невеликих відхилень початкових даних. [14]

На відміну від більшості інших інтерполяційних методів, спосіб інтерполяційної Гаус-функції, достатньо легко застосовується на n -вимірний простір, наприклад, тривимірний, для інтерполяції неупорядкованої множини дискретних даних.

В запропонованому варіанті є мінуси, спільні для методів з явним визначенням функцій інтерполяції. Наприклад, за допомогою такого способу не можуть бути описані криві і поверхні. [69] Проведемо параметризацію інтерполяційної функції Гауса, для того, щоб усунути цей недолік.

Таким чином, після отримання параметричних рівнянь (2.4) і (2.5) та підстановки у ці функції параметру t , отримаємо значення x та y , які будуть утворювати інтерполяційну криву Гауса у параметричному вигляді. [12]

Використання параметричної функції Гауса дозволяє проводити інтерполяцію замкнутого каркасу точок. Це доречно в умовах сформульованої в дипломній роботі задачі, де необхідно побудувати замкнений полігон ураження лісу пожежею. [17] Приклад вирішення такої задачі зображений на рисунку 2.3.

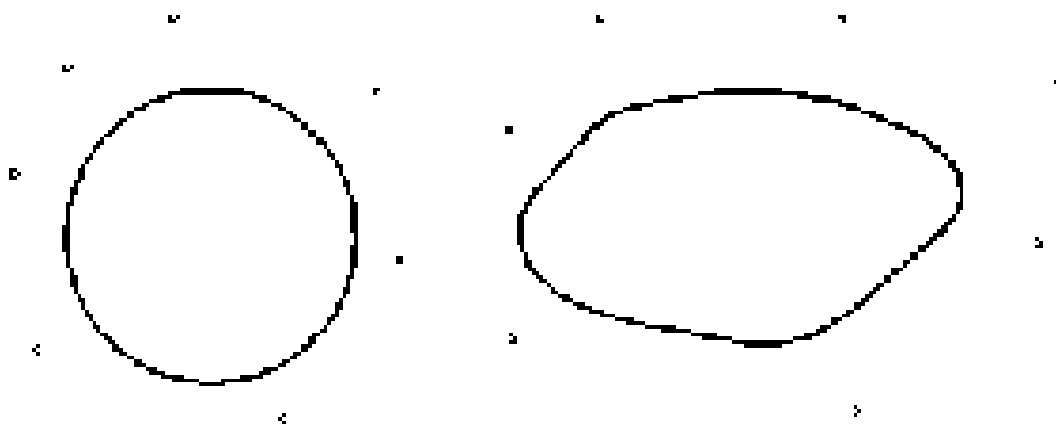


Рисунок 2.3. — Інтерполяція замкненого каркасу із застосуванням параметричної функції Гауса

2.2. Методи деформаційного моделювання

Метод скінчених елементів. Відомим методом для моделювання деформованих об'єктів є метод скінчених елементів, який широко впроваджується в інженерній роботі, наприклад, використовується для моделювання напруг в будівельних конструкціях. [16] Метод включає декомпозицію об'єкта на складові з базисних елементів та може використовуватись для двовимірного та тривимірного випадків однакового чином, в залежності від вибору елемента. Кожен елемент складається з ряду вузлів, визначаючи форму елемента. [70] Існує декілька типів

елементів, які можуть використовуватись для тривимірного випадку: куб, призма, піраміда, тетраедр, тощо, як показано на рис. 2. 4. [19]

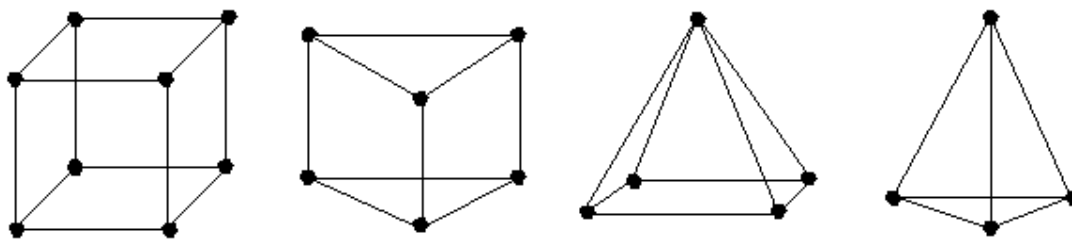


Рис. 2.4. Лінійні тривимірні скінченні елементи

Тут представлено лінійні елементи, кожне ребро яких визначається лише двома вузлами. Це означає, що при будь-яких деформаціях елемента кожне ребро залишиться прямою лінією (рис. 2.5). [20]

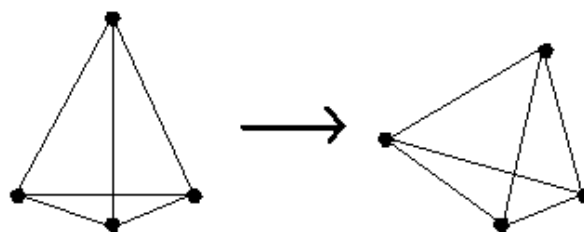


Рис. 2.5. Деформація лінійного тетраедрального скінченного елемента

Існують також квадратичні елементи з додатковим вузлом на кожному ребрі, та кубічні елементи, з двома додатковими вузлами (рис. 2.6).

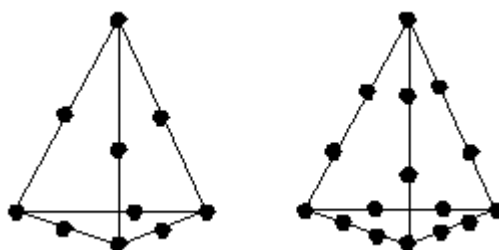


Рис. 2.6. Квадратичні та кубічні скінченні елементи

Грані елементів більш високого порядку мають більше ніж два кінцевих вузла, дозволяючи деформацію граней. Це також дає більшу точність відображення на кожен елемент. [21] Платою за це є збільшення обчислювальних затрат, тому що кожен новий вузол додає нове невідоме у систему рівнянь. Тут потрібен компроміс між точністю та часом моделювання.

Елементи зв'язують разом через спільне використання вузлів між суміжними елементами. Деформація об'єкта обчислюється статично мінімізацією рівняння статичної рівноваги по всіх елементах:

$$\mathbf{W_r} = \mathbf{W_b} + \mathbf{W_s} + \mathbf{W_f}, \quad (2.5)$$

де: $\mathbf{W_r}$ - внутрішня робота, що виконується завдяки внутрішнім напругам;

$\mathbf{W_b}$ - робота завдяки силам тіла (наприклад, силі тяжіння);

$\mathbf{W_s}$ - робота завдяки поверхневим зусиллям (наприклад, тиску);

$\mathbf{W_f}$ - робота, що виконується дотичними до вузлів силами.

Метод використовує мінімізацію рівняння енергії в матричній формі:

$$\mathbf{K} = \mathbf{U} \mathbf{R}, \quad (2.6)$$

де: \mathbf{K} - матриця "жорсткостей" для об'єкта,

\mathbf{U} - вектор, що містить всі вузлові зміщення,

\mathbf{R} - вектор, що містить всі прикладені сили.

Оскільки \mathbf{K} відомо, а \mathbf{R} може бути обчислено, \mathbf{U} , тобто позиція усіх вузлів, також може бути знайдена. Перевага методу полягає в його точності. Однак велика кількість рівнянь у системі, яку потрібно розв'язати, робить цей метод складним для використання у реальному часі. [23] Також модель не має справи з нерезидентними напругами (наприклад, від зіткнення). Для цього мають бути додані інерція та фактори демпфування, а також швидкість зміщення та прискорення.

Метод TLM. Перевага методу TLM над методом скінчених елементів полягає в його здатності до моделювання нерезидентних напруг, так як цей метод моделює хвилі напруги в об'єкті. Як і з методом скінчених елементів, обчислюються вузлові

зміщення, які призводять до деформації об'єкта. Ефективність TLM з постійними напругами та великими деформаціями ще недостатньо досліджена на предмет швидкості розв'язку в задачах реального часу. [18]

Метод довільної деформації форми. Загальна геометрична процедура для моделювання процесів деформації твердотільних об'єктів отримала назву методу довільної деформації форми, або, скорочено, FFD (від англ. Free-Form Deformation). Метод FFD вперше було представлено Томасом Седербергом та Скоттом Перрі у роботі [24], причому поняття деформації визначалося як зміна форми об'єкта під впливом зовнішніх сил та внутрішніх напружень, а сам FFD – як просторово-орієнтований метод деформації полігональних (або визначених множиною контрольних точок) геометричних об'єктів. [25]

Основна ідея методу FFD полягає у зануренні геометричної моделі або тієї її частини, яка має бути деформована, у тривимірну решітку довільної форми. Ця решітка може розглядатися як деякий паралелепіпед з прозорого гнучкого пластика, який охоплює цю модель. Таким чином, деформуючи решітку, ми “автоматично” деформуємо модель. Математично процедура FFD може бути описана в термінах тензорного числення як знаходження тензорного добутку тривимірних поліномів Бернштейна [26, 27].

Нехай в результаті деформації контрольні точки $\mathbf{F}_{i,j,k}^{pp}$ паралелепіпеда переходять у нові точки $\mathbf{F}_{i,j,k}$. Тоді нове положення будь-якої точки об'єкта може бути обчислено шляхом визначення її локальних

координат за формулою:

$$\mathbf{F}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n \mathbf{F}_{i,j,k} \mathbf{B}_i^l(\mathbf{u}) \mathbf{B}_j^m(\mathbf{v}) \mathbf{B}_k^n(\mathbf{w}). \quad (2.7)$$

де $\mathbf{u}, \mathbf{v}, \mathbf{w}$ – локальні координати точки об'єкта, які визначаються обраною для деформації решіткою розмірністю $\mathbf{l} \times \mathbf{m} \times \mathbf{n}$ точок, а \mathbf{B} – коефіцієнти Безьє. [71]

Таким чином, точка \mathbf{P} об'єкта внаслідок деформації переходить у точку $\mathbf{F}(\mathbf{u}, \mathbf{v}, \mathbf{w})$. Метод FFD дозволяє моделювати деформації об'єктів будь-якого характеру: площини, поверхні другого порядку, порції параметрично заданих поверхонь або неявно заданих поверхонь. [72] Процедура деформації може проводитися як глобально (для всієї моделі), так і локально (для частини моделі); крім того, що дуже важливо, є можливість задання умов збереження об'єму об'єкта, що моделюється. Використання методу FFD дає можливість будувати фізично обумовлені моделі деформацій завдяки тому, що моделювання деформації контрольної решітки, як правило, може бути виконано безпосередньо за формулами з опору матеріалів, а потім “автоматично” перенесено на реальний об'єкт. [28]

Але метод FFD має також ряд недоліків, одним з яких для “класичного” варіанту методу є те, що користувачі, які не знайомі зі сплайнами, погано розуміють сутність контрольних точок та результати їх переміщень, при програмній реалізації методу точки контрольної решітки змішуються на екрані з точками об'єкта, засмічують сцену та ускладнюють роботу. [22]

“Класичний” варіант методу довільної деформації форми служить основою для багатьох похідних геометричних методів деформативного моделювання. Так, у роботі [30] пропонується високоінтерактивна та інтуїтивно зрозуміла техніка моделювання на базі FFD, що дозволяє проводити деформацію поверхонь шляхом введення збурень довільної форми або згибу вздовж довільної кривої. В цьому методі, названому EFFD (від англійського Extended FFD), було вперше зроблено спробу скористатися просторовими решітками з формами, відмінними від паралелепіпеда, завдяки яким визначається зовнішній вигляд деформації. Ті ж автори у роботі [31] пропонують техніку AFFD (від англ. Animated FFD), яка дозволяє моделювати динаміку деформації завдяки залученню методу анімації з використанням ключових кадрів (keyframing animation).

Метод FFD також може бути узагальненим завдяки використанню інтерполяції для розсіяних вихідних даних за допомогою тріангуляції Делоне та діаграм Вороного-Діріхле [29]. Цей метод отримав назву DFFD (від франц. Dirichlet

FFD). До його переваг слід віднести, по-перше, значне спрощення контролю локальних деформацій, і, по-друге, його сумісність з усіма іншими похідними FFD.

В базовому методі FFD завдяки використанню стратегії послідовних розділень Кетмулла-Кларка можуть бути використані решітки довільної форми (топології) [35]. Це знімає найбільш суттєве обмеження на використання „класичного“ варіанта методу, а саме необхідність використання контрольного каркасу у вигляді паралелограма, що неприйнятно для об'єктів складної конфігурації внаслідок великих похибок при моделюванні.

Зважаючи на складність та багатоплановість проблем, що виникають при геометричному моделюванні деформацій, доцільно було б залучити для їх розв'язання методи деформативного моделювання на базі геометричних (біраціональних, політканинних та інших) перетворень. [34]

3. МАТЕМАТИЧНИЙ АПАРАТ СИСТЕМИ МОДЕЛЮВАННЯ ПРОЦЕСІВ ДЕФОРМАЦІЇ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ

У даному розділі розглянуто математичний апарат системи моделювання процесів деформації геометричних об'єктів на основі політочкових перетворень.

У даному підрозділі розділі розглянуто спосіб деформаційного моделювання об'єктів за допомогою політочкових перетворень, розглянуті різні варіанти функціоналів цих перетворень, а також проаналізовано апарат вагових функцій, наводиться удосконалення методу політочкових перетворень, яке зменшує похибку обчислень при прогнозуванні. В розділі наводиться методика розрахунку збитків від наслідків лісової пожежі. [33]

Політочкові перетворення у конструюванні деформації об'єктів. Для початку необхідно розглянути політочкові перетворення з двома точками базису. Розглянемо політочкові перетворення, базисом яких є множина точок, а об'єктом - пряма.

Визначимо однорідний простір:

$$\mathbf{X}_{\text{однор.}} = \mathbf{W}\mathbf{X},$$

$$\mathbf{Y}_{\text{однор.}} = \mathbf{W}\mathbf{Y},$$

$$\mathbf{Z}_{\text{однор.}} = \mathbf{W}.$$

В однорідному просторі рівняння прямої буде мати вигляд:

$$\mathbf{a} \mathbf{X}_{\text{однор.}} + \mathbf{b} \mathbf{Y}_{\text{однор.}} + \mathbf{c} \mathbf{Z}_{\text{однор.}} = 0.$$

Це рівняння є симетричним відносно координат $\mathbf{X}_{\text{однор.}}$, $\mathbf{Y}_{\text{однор.}}$, $\mathbf{Z}_{\text{однор.}}$ та \mathbf{a} , \mathbf{b} , \mathbf{c} . Тобто, можна сказати, що пряма визначається координатами \mathbf{a} , \mathbf{b} , \mathbf{c} , а будь-яка точка простору – координатами $\mathbf{X}_{\text{однор.}}$, $\mathbf{Y}_{\text{однор.}}$, $\mathbf{Z}_{\text{однор.}}$. [36]

При двоточкових перетвореннях задається дві точки каркасу і пряма (прообраз). [37] Користувач змінює точки каркасу та за допомогою політочкових перетворень отримує нову пряму (прообраз).

Розглянемо геометричну ілюстрацію такої задачі (рисунок 3.1.).

Дві точки каркасу O_1'' та O_2'' задані декартовими координатами x_1'', y_1'' та x_2'', y_2'' . Проводиться висхідна пряма (пряма, яка підлягає перетворенню) і визначається відстань від цих точок до заданої прямої. [51]

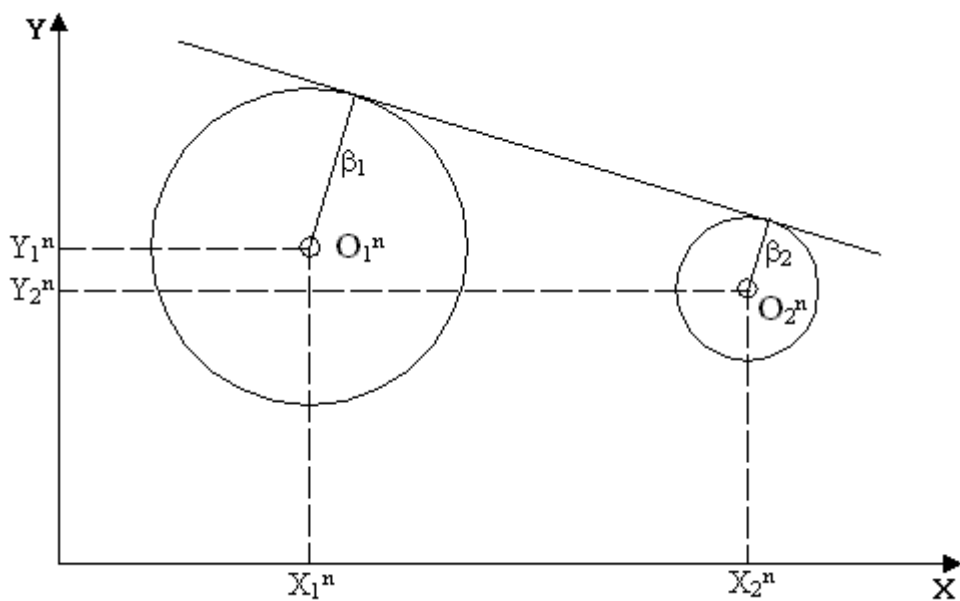
Для цього необхідно нормалізувати рівняння прямої (за умови **Zоднор.** =1), тобто поділити коефіцієнти на

$$\sqrt{a^2 + b^2}$$

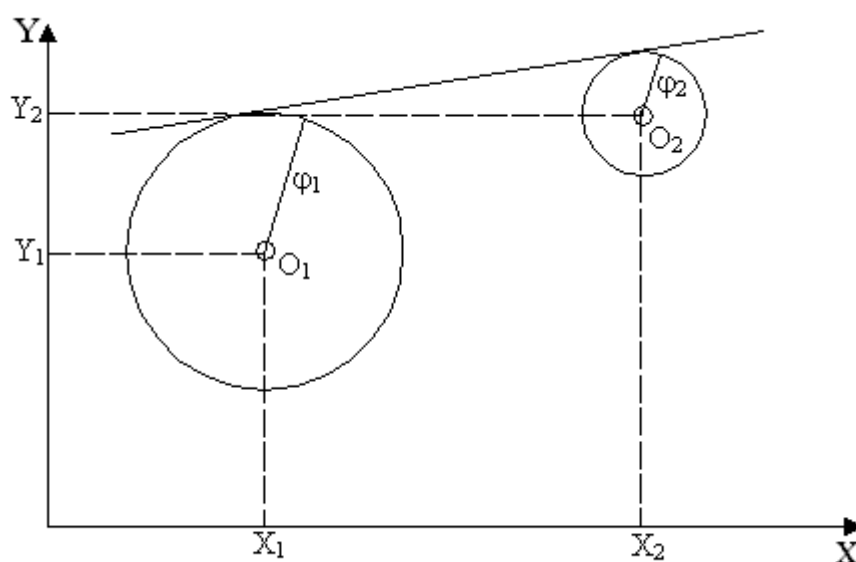
з урахуванням знаку(+,-), який є протилежним до **c**. Після підстановки x_1'', y_1'' та x_2'', y_2'' у нормалізоване рівняння отримуються координати β_1 та β_2 , які є полікоординатами названих точок. [38]

Тобто, маємо:

1. 2 точки O_1'' та O_2'' , визначаються декартовими координатами (x_1'', y_1'') та (x_2'', y_2'') та політочковими координатами β_1 та β_2 ;
2. пряму - прообраз, яка є дотичною до двох кіл з радіусами β_1 та β_2 і з центрами O_1'' та O_2'' .



а) Пряма лінія у початковому базисі



б) Пряма лінія у перетвореному базисі

Рисунок 3.1. — Політочкові перетворення при 2 точках базису

Змінюється каркас. Нові точки позначено O_1 та O_2 (рисунок 3.1,б), а відповідні їм нові координати (x_1, y_1) і (x_2, y_2) . [39] При цьому політочкові координати не змінилися.

Перетворену пряму шукають у вигляді:

$$Ax + By + C = 0.$$

Для пошуку коефіцієнтів **A, B, C**, потрібно розв'язати систему рівнянь такого вигляду:

$$\begin{cases} Ax_1 + By_1 + C = \beta_1, \\ Ax_2 + By_2 + C = \beta_2, \\ A^2 + B^2 = 1, \end{cases} \quad (3.1)$$

де x_1, y_1, x_2, y_2 - декартові координати перетвореного базису, β_1 та β_2 - політочкові координати заданої прямої, а **A, B, C** - коефіцієнти перетвореної прямої, які необхідно визначити. [41]

Розв'яжемо цю систему.

Від 1-го рівняння віднімаємо 2-ге:

$$A(x_1 - x_2) + B(y_1 - y_2) = \beta_1 - \beta_2.$$

Позначимо $x = x_1 - x_2$; $y = y_1 - y_2$, $d = \beta_1 - \beta_2$.

Тоді:

$$\begin{cases} Ax + By = d, \\ A = \sqrt{1 - B^2}. \end{cases}$$

Підставляємо **A** в перше рівняння:

$$\sqrt{1 - B^2} \cdot x + By = d,$$

$$(1 - B^2) \cdot x^2 = (d - By)^2,$$

$$x^2 - x^2 B^2 = d^2 - 2 dBy + B^2 y^2,$$

$$(y^2 + x^2) \cdot B^2 - 2 dyB + d^2 - x^2 = 0.$$

Отримали квадратне рівняння, яке треба розв'язати:

$$D = 4 d^2 y^2 - 4(y^2 + x^2)(d^2 - x^2) = 4(d^2 y^2 - y^2 d^2 - x^2 d^2 + x^4 + x^2 y^2) = 4x^2(x^2 - d^2 + y^2).$$

$$B_1 = \frac{2dy - \sqrt{4x^2(x^2 - d^2 + y^2)}}{2(y^2 + x^2)} = \frac{dy - x\sqrt{x^2 - d^2 + y^2}}{x^2 + y^2},$$

$$B_2 = \frac{dy + x\sqrt{x^2 - d^2 + y^2}}{x^2 + y^2}.$$

Коефіцієнти A підрахуємо за допомогою визначених коефіцієнтів B :

$$A = \pm\sqrt{1 - B^2},$$

і таких коефіцієнтів буде 4: $A_{1,1}$, $A_{1,2}$, $A_{2,1}$, $A_{2,2}$, а саме:

$$A_{1,1} = \sqrt{1 - B_1^2},$$

$$A_{1,2} = -\sqrt{1 - B_1^2},$$

$$A_{2,1} = \sqrt{1 - B_2^2},$$

$$A_{2,2} = -\sqrt{1 - B_2^2}.$$

Оскільки отримано 4 коефіцієнти $A_{1,1}$, $A_{1,2}$, $A_{2,1}$, $A_{2,2}$, то це свідчить про те, що розв'язком системи (3.1) будуть 4 прямі, тобто розв'язок не буде однозначним (рисунок 3.2). [40] Тому потрібно знайти пряму, яка знаходиться на певній відстані від заданих точок нового базису.

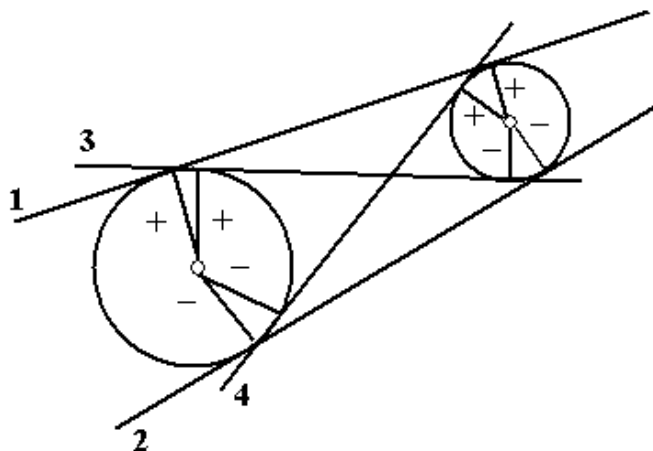


Рисунок 3.2. — Можливе розташування дотичних до двох кіл з урахуванням знаку відстаней

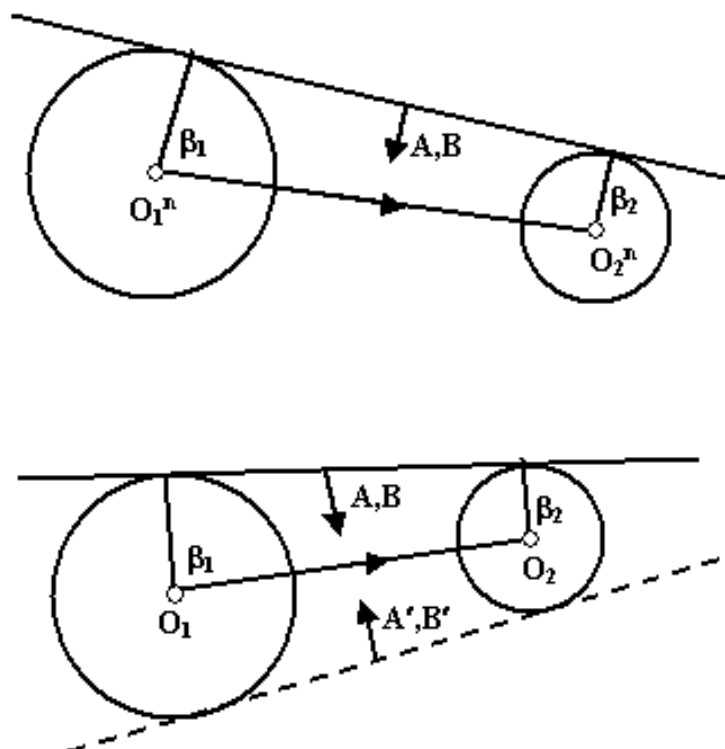


Рисунок 3.3. — Позбавлення зайвої прямої - розв'язку системи при поліоточкових перетвореннях

Математично процедуру вибору однієї з двох можливих прямих [52] при поліоточкових перетвореннях можна записати так (рисунок 3.3):

$$\text{знак } (\overrightarrow{O_1^n O_2^n}) \times (\overrightarrow{a, b}) = \text{знак } (\overrightarrow{O_1 O_2}) \times (\overrightarrow{A, B}),$$

$$\text{знак } (\overrightarrow{O_1^n O_2^n}) \times (\overrightarrow{a, b}) \neq \text{знак } (\overrightarrow{O_1 O_2}) \times (\overrightarrow{A', B'}),$$

що не підходить, тобто векторний добуток двох векторів дає знак “плюс”, якщо перший вектор повертається до другого в тому напрямку, що і вісь X до вісі Y , і “мінус”, якщо навпаки. Користуючись вказаною правилом, можна позбавитись зайвої прямої. [42]

У випадку, коли каркас двоточковий, можна вимагати, щоб при перетворенні не змінювались координати β_1 і β_2 . При багатоточковому каркасі це неможливо.

Поліоточкові перетворення при багатоточковому базисі. [73] Початковий базис в однорідному просторі задається базисними точками з координатами

$$x_i'', y_i'', z_i'', i=1,2,...,p.$$

Пряма (прообраз) визначається за допомогою таких рівнянь

$$ax_i'' + by_i'' + cz_i'' = \beta_i, \quad i = 1, ..., p,$$

де a, b, c – координати прямої (прообразу).

Політочкові координати β_i – відстань зі своїм знаком від точок до прямої(образу). [74] Перетворений базис буде заданий у вигляді x_i, y_i, z_i . Нові координати будуть визначатись за формулами:

$$\varphi_i = Ax_i + By_i + Cz_i, \quad (3.2)$$

де x_i, y_i, z_i – координати нового базису, A, B, C – невідомі коефіцієнти нової прямої. Значення β_i та φ_i називаються політочковими координатами прямих (рисунок 3.4).

У випадку багатоточкового каркасу будуть змінюватись і декартові координати x_i, y_i, z_i , й політочкові координати β_i прямої. [75] Вони отримають нові значення φ_i .

У загальному випадку $\beta_i \neq \varphi_i, i = 1, 2, ..., p$.

Політочкові перетворення можна записати у вигляді:

$$\varphi_i = \omega_i \beta_i, \quad i = 1, 2, ..., p.$$

Підставимо замість φ_i в ліву частину рівняння (3.1) $\omega_i \beta_i$:

$$\omega_i \beta_i = Ax_i + By_i + Cz_i, \quad i = 1, 2, ..., p.$$

Ця система містить p рівнянь та $p+3$ невідомих ($\omega_i, i = 1, 2, ..., p, A, B, C$).

Таким чином, для отримання однозначного розв'язку системи вводяться три додаткові умови.

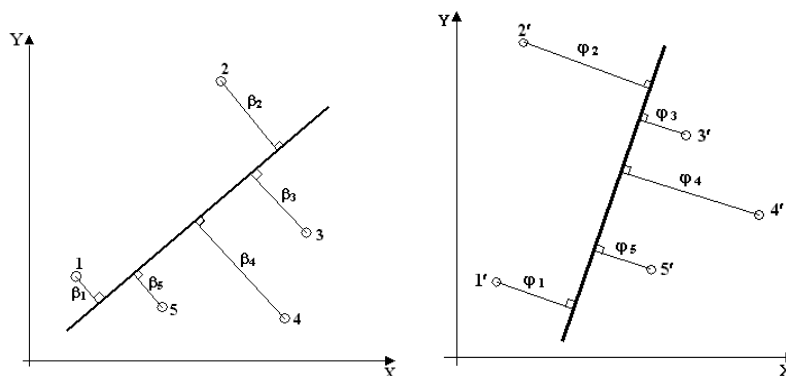


Рисунок 3.4. — Політочкові перетворення при багатоточковому базисі

Для отримання додаткових умов пропонується знайти ω_i , $i = 1, 2, \dots, p$ за умови їх мінімального відхилення від одиниці, тоді φ_i будуть мало відхилятися від β_i . [43] Таким чином, пропонується мінімізувати функціонал такого вигляду:

$$S = \sum_{i=1}^p (\omega_i - 1)^2 \rightarrow \min.$$

Підставимо замість ω_i вираз $\frac{\varphi_i}{\beta_i}$:

$$S = \sum_{i=1}^p \left(\frac{\varphi_i}{\beta_i} - 1 \right)^2 = \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + CZ_i)}{\beta_i} \right) - 1 \right)^2 \rightarrow \min.$$

Необхідно знайти **A**, **B**, **C**, які є розв'язком системи, для чого знаходимо частинні похідні за **A** та **B**. [76] Третє рівняння системи впливає з умови, що отримана пряма повинна бути нормалізованою.

$$\frac{\partial S}{\partial A} = 2 \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + CZ_i)}{\beta_i} \right) - 1 \right) \left(\frac{X_i}{\beta_i} \right) = 0;$$

$$\frac{\partial S}{\partial B} = 2 \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + CZ_i)}{\beta_i} \right) - 1 \right) \left(\frac{Y_i}{\beta_i} \right) = 0;$$

Розкриємо дужки:

$$A \sum_{i=1}^p \frac{(X_i X_i)}{\beta_i^2} + B \sum_{i=1}^p \frac{(X_i Y_i)}{\beta_i^2} + C \sum_{i=1}^p \frac{(X_i Z_i)}{\beta_i^2} - \sum_{i=1}^p \frac{X_i}{\beta_i} = 0;$$

$$A \sum_{i=1}^p \frac{(Y_i X_i)}{\beta_i^2} + B \sum_{i=1}^p \frac{(Y_i Y_i)}{\beta_i^2} + C \sum_{i=1}^p \frac{(Y_i Z_i)}{\beta_i^2} - \sum_{i=1}^p \frac{Y_i}{\beta_i} = 0;$$

Система, розв'язок якої дасть нам коефіцієнти перетвореної прямої, буде мати вигляд:

$$\left\{ \begin{array}{l} A \sum_{i=1}^p \frac{X_i^2}{\beta_i^2} + B \sum_{i=1}^p \frac{X_i Y_i}{\beta_i^2} + C \sum_{i=1}^p \frac{X_i Z_i}{\beta_i^2} = \sum_{i=1}^p \frac{X_i}{\beta_i}, \\ A \sum_{i=1}^p \frac{X_i Y_i}{\beta_i^2} + B \sum_{i=1}^p \frac{Y_i^2}{\beta_i^2} + C \sum_{i=1}^p \frac{Y_i Z_i}{\beta_i^2} = \sum_{i=1}^p \frac{Y_i}{\beta_i}, \\ A^2 + B^2 = 1. \end{array} \right.$$

Розділимо перше та друге рівняння системи на вирази, які стоять поряд з C для того, щоб коефіцієнт біля C дорівнював одиниці. [53] Тоді маємо:

$$\left\{ \begin{array}{l} A \frac{\sum_{i=1}^p \frac{X_i}{\beta_i^2}}{\sum_{i=1}^p \frac{X_i Z_i}{\beta_i^2}} + B \frac{\sum_{i=1}^p \frac{X_i Y_i}{\beta_i^2}}{\sum_{i=1}^p \frac{X_i Z_i}{\beta_i^2}} + C = \frac{\sum_{i=1}^p \frac{X_i}{\beta_i}}{\sum_{i=1}^p \frac{X_i Z_i}{\beta_i^2}}, \\ A \frac{\sum_{i=1}^p \frac{X_i Y_i}{\beta_i^2}}{\sum_{i=1}^p \frac{Y_i Z_i}{\beta_i^2}} + B \frac{\sum_{i=1}^p \frac{Y_i^2}{\beta_i^2}}{\sum_{i=1}^p \frac{Y_i Z_i}{\beta_i^2}} + C = \frac{\sum_{i=1}^p \frac{Y_i}{\beta_i}}{\sum_{i=1}^p \frac{Y_i Z_i}{\beta_i^2}}, \\ A^2 + B^2 = 1. \end{array} \right.$$

У цій системі X_i, Y_i, Z_i - декартові координати точок нового базису, β_i - політочкові координати заданої прямої. Коли ми підставимо їх у систему, то отримаємо біля A і B числові значення. [44] Таким чином, отримана система співпадає за виглядом з системою (3.1) і розв'язується аналогічно до неї. Розв'язком системи будуть координати A, B та C , тобто нова пряма $AX + BY + C = 0$.

Для визначення точки можна використати дві прямі, які перетинаються, наприклад:

$$\begin{cases} \mathbf{a}_1 \mathbf{X} + \mathbf{c}_1 = 0, \\ \mathbf{b}_2 \mathbf{Y} + \mathbf{c}_2 = 0. \end{cases}$$

Тоді:

$$\begin{cases} \mathbf{X} = -\frac{\mathbf{c}_1}{\mathbf{a}_1}, \\ \mathbf{Y} = -\frac{\mathbf{c}_2}{\mathbf{b}_2}, \end{cases}$$

і надалі можна працювати вже з цими прямими.

Якщо прирівняти $\mathbf{z}=1$, у випадку \mathbf{n} точок базису система буде мати вигляд:

$$\begin{cases} \mathbf{A}\mathbf{x}_{10} + \mathbf{B}\mathbf{y}_{10} + \mathbf{C} = \mathbf{d}_{10}, \\ \mathbf{A}\mathbf{x}_{20} + \mathbf{B}\mathbf{y}_{20} + \mathbf{C} = \mathbf{d}_{20}, \\ \mathbf{A}^2 + \mathbf{B}^2 = 1, \end{cases}$$

де

$$\begin{aligned} \mathbf{x}_{10} &= \frac{\sum_{i=1}^p \frac{\mathbf{x}_i^2}{\beta_i^2}}{\sum_{i=1}^p \frac{\mathbf{x}_i}{\beta_i^2}}, \quad \mathbf{y}_{10} = \frac{\sum_{i=1}^p \frac{\mathbf{x}_i \mathbf{y}_i}{\beta_i^2}}{\sum_{i=1}^p \frac{\mathbf{x}_i}{\beta_i^2}}, \quad \mathbf{d}_{10} = \frac{\sum_{i=1}^p \frac{\mathbf{x}_i}{\beta_i}}{\sum_{i=1}^p \frac{\mathbf{x}_i}{\beta_i^2}}, \\ \mathbf{x}_{20} &= \frac{\sum_{i=1}^p \frac{\mathbf{x}_i \mathbf{y}_i}{\beta_i^2}}{\sum_{i=1}^p \frac{\mathbf{y}_i}{\beta_i^2}}, \quad \mathbf{y}_{20} = \frac{\sum_{i=1}^p \frac{\mathbf{y}_i^2}{\beta_i^2}}{\sum_{i=1}^p \frac{\mathbf{y}_i}{\beta_i^2}}, \quad \mathbf{d}_{20} = \frac{\sum_{i=1}^p \frac{\mathbf{y}_i}{\beta_i}}{\sum_{i=1}^p \frac{\mathbf{y}_i}{\beta_i^2}}, \end{aligned}$$

де $\mathbf{x}_i, \mathbf{y}_i$ - декартові координати точок базису, а β_i – відстань від точок базису до прямої. [45] Ця система співпадає за виглядом з системою (3.1) і розв'язується аналогічно до неї. Розв'язком системи будуть \mathbf{A}, \mathbf{B} , та \mathbf{C} - коефіцієнти нової прямої.

Визначення коефіцієнтів прямої при двоточкових перетвореннях. У найпростішому випадку, коли базис має дві точки, для знаходження нової прямої із застосуванням мінімізуючого функціоналу необхідно розв'язати систему такого вигляду:

$$\begin{cases} Ax_{12} + By_{12} + C = d_{12}, \\ Ax_{22} + By_{22} + C = d_{22}, \\ A^2 + B^2 = 1, \end{cases} \quad (3.3)$$

де

$$\begin{aligned} x_{12} &= \frac{\frac{x_1^2}{\beta_1^2} + \frac{x_2^2}{\beta_2^2}}{\frac{x_1}{\beta_1^2} + \frac{x_2}{\beta_2^2}}, & x_{22} &= \frac{\frac{x_1 y_1}{\beta_1^2} + \frac{x_2 y_2}{\beta_2^2}}{\frac{y_1}{\beta_1^2} + \frac{y_2}{\beta_2^2}}, \\ y_{12} &= \frac{\frac{x_1 y_1}{\beta_1^2} + \frac{x_2 y_2}{\beta_2^2}}{\frac{x_1}{\beta_1^2} + \frac{x_2}{\beta_2^2}}, & y_{22} &= \frac{\frac{y_1^2}{\beta_1^2} + \frac{y_2^2}{\beta_2^2}}{\frac{y_1}{\beta_1^2} + \frac{y_2}{\beta_2^2}}, \\ d_{12} &= \frac{\frac{x_1 \beta_1}{\beta_1^2} + \frac{x_2 \beta_2}{\beta_2^2}}{\frac{x_1}{\beta_1^2} + \frac{x_2}{\beta_2^2}}, & d_{22} &= \frac{\frac{y_1 \beta_1}{\beta_1^2} + \frac{y_2 \beta_2}{\beta_2^2}}{\frac{y_1}{\beta_1^2} + \frac{y_2}{\beta_2^2}}. \end{aligned}$$

Розв'язок цієї системи рівнянь (у випадку двох точок базису) буде співпадати з розв'язком спрощеної системи вигляду:

$$\begin{cases} Ax_1 + By_1 + C = d_1, \\ Ax_2 + By_2 + C = d_2, \\ A^2 + B^2 = 1, \end{cases} \quad (3.4)$$

Твердження 1. У випадку двоточкових перетворень розв'язки систем із застосуванням оптимізаційного функціоналу (3.3) та без нього (3.4) збігаються.

Доведення. Для розв'язання спрощеної системи вводилися позначення $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$, $\mathbf{y} = \mathbf{y}_1 - \mathbf{y}_2$, $\mathbf{d} = \mathbf{d}_1 - \mathbf{d}_2$, і через ці позначення виводилися формули для визначення коефіцієнтів прямої \mathbf{A} , \mathbf{B} та \mathbf{C} . А саме:

$$\mathbf{B} = \frac{\left(\mathbf{y}\mathbf{d} \pm \sqrt{\mathbf{x}^2 + \mathbf{y}^2 - \mathbf{d}^2}\right)}{\left(\mathbf{x}^2 + \mathbf{y}^2\right)},$$

$$\mathbf{A} = \sqrt{1 - \mathbf{B}^2},$$

$$\mathbf{C} = \frac{(\mathbf{d} - \mathbf{B}\mathbf{y})}{\mathbf{x}}.$$

Знайдемо, чому дорівнюють $\mathbf{x} = \mathbf{x}_{12} - \mathbf{x}_{22}$, $\mathbf{y} = \mathbf{y}_{12} - \mathbf{y}_{22}$, $\mathbf{d} = \mathbf{d}_{12} - \mathbf{d}_{22}$ для системи (3.3). Для прикладу візьмемо $\mathbf{y} = \mathbf{y}_{12} - \mathbf{y}_{22}$:

$$\begin{aligned} \mathbf{y} = \mathbf{y}_{12} - \mathbf{y}_{22} &= \frac{\frac{\mathbf{x}_1\mathbf{y}_1}{\beta_1^2} + \frac{\mathbf{x}_2\mathbf{y}_2}{\beta_2^2}}{\frac{\mathbf{x}_1}{\beta_1^2} + \frac{\mathbf{x}_2}{\beta_2^2}} - \frac{\frac{\mathbf{y}_1^2}{\beta_1^2} + \frac{\mathbf{y}_2^2}{\beta_2^2}}{\frac{\mathbf{y}_1}{\beta_1^2} + \frac{\mathbf{y}_2}{\beta_2^2}} = \\ &= \frac{\mathbf{x}_1\mathbf{y}_1\beta_2^2 + \mathbf{x}_2\mathbf{y}_2\beta_1^2}{\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2} - \frac{\mathbf{y}_1^2\beta_2^2 + \mathbf{y}_2^2\beta_1^2}{\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2} = \\ &= \frac{(\mathbf{x}_1\mathbf{y}_1\beta_2^2 + \mathbf{x}_2\mathbf{y}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)}{(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)} - \\ &\quad - \frac{(\mathbf{y}_1^2\beta_2^2 + \mathbf{y}_2^2\beta_1^2)(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)}{(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)} = \\ &= \frac{\mathbf{x}_1\mathbf{y}_1^2\beta_2^4 + \mathbf{x}_2\mathbf{y}_2\mathbf{y}_1\beta_1^2\beta_2^2 + \mathbf{x}_1\mathbf{y}_2\mathbf{y}_1\beta_1^2\beta_2^2 + \mathbf{x}_2\mathbf{y}_2^2\beta_1^4}{(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)} + \\ &\quad + \frac{-\mathbf{x}_1\mathbf{y}_1^2\beta_2^4 - \mathbf{x}_1\mathbf{y}_2^2\beta_1^2\beta_2^2 - \mathbf{x}_2\mathbf{y}_1^2\beta_1^2\beta_2^2 - \mathbf{x}_2\mathbf{y}_2^2\beta_1^4}{(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)} = \\ &= (\mathbf{y}_1 - \mathbf{y}_2) \frac{\beta_1^2\beta_2^2(\mathbf{x}_1\mathbf{y}_2 - \mathbf{x}_2\mathbf{y}_1)}{(\mathbf{x}_1\beta_2^2 + \mathbf{x}_2\beta_1^2)(\mathbf{y}_1\beta_2^2 + \mathbf{y}_2\beta_1^2)}. \end{aligned}$$

Аналогічно отримуємо вирази для $\mathbf{x} = \mathbf{x}_{12} - \mathbf{x}_{22}$ та $\mathbf{d} = \mathbf{d}_{12} - \mathbf{d}_{22}$.

$$\mathbf{x} = \mathbf{x}_{12} - \mathbf{x}_{22} = (\mathbf{x}_1 - \mathbf{x}_2) \frac{\beta_1^2 \beta_2^2 (\mathbf{x}_1 y_2 - \mathbf{x}_2 y_1)}{(\mathbf{x}_1 \beta_2^2 + \mathbf{x}_2 \beta_1^2)(y_1 \beta_2^2 + y_2 \beta_1^2)},$$

$$\mathbf{d} = \beta_{12} - \beta_{22} = (\beta_1 - \beta_2) \frac{\beta_1^2 \beta_2^2 (\mathbf{x}_1 y_2 - \mathbf{x}_2 y_1)}{(\mathbf{x}_1 \beta_2^2 + \mathbf{x}_2 \beta_1^2)(y_1 \beta_2^2 + y_2 \beta_1^2)}.$$

Нехай \mathbf{R} - деякий множник, який визначається за формулою:

$$\mathbf{R} = \frac{\beta_1^2 \beta_2^2 (\mathbf{x}_1 y_2 - \mathbf{x}_2 y_1)}{(\mathbf{x}_1 \beta_2^2 + \mathbf{x}_2 \beta_1^2)(y_1 \beta_2^2 + y_2 \beta_1^2)}.$$

Тоді можна записати :

$$\mathbf{x} = \mathbf{x}_{12} - \mathbf{x}_{22} = (\mathbf{x}_1 - \mathbf{x}_2) \mathbf{R},$$

$$\mathbf{y} = \mathbf{y}_{12} - \mathbf{y}_{22} = (\mathbf{y}_1 - \mathbf{y}_2) \mathbf{R},$$

$$\mathbf{d} = \beta_{12} - \beta_{22} = (\beta_1 - \beta_2) \mathbf{R},$$

$$\mathbf{R} = \frac{\mathbf{d}_1^2 \mathbf{d}_2^2 (\mathbf{x}_1 y_2 - \mathbf{x}_2 y_1)}{(\mathbf{x}_1 \mathbf{d}_2^2 + \mathbf{x}_2 \mathbf{d}_1^2)(y_1 \mathbf{d}_2^2 + y_2 \mathbf{d}_1^2)}.$$

Якщо підставити отримані результати у формулу для обчислення коефіцієнтів нової прямої, отримаємо:

$$\begin{aligned} \mathbf{B} &= \frac{\left(\mathbf{y} \mathbf{R} \mathbf{d} \mathbf{R} \pm \sqrt{\mathbf{R}^2 \mathbf{x}^2 + \mathbf{R}^2 \mathbf{y}^2 - \mathbf{R}^2 \mathbf{d}^2} \right)}{\mathbf{R}^2 (\mathbf{x}^2 + \mathbf{y}^2)} = \\ &= \frac{\left(\mathbf{y} \mathbf{d} \pm \sqrt{\mathbf{x}^2 + \mathbf{y}^2 - \mathbf{d}^2} \right)}{(\mathbf{x}^2 + \mathbf{y}^2)}. \end{aligned}$$

Таким чином коефіцієнт \mathbf{B} буде визначатись за тією ж формулою, що і при розв'язуванні спрощеної системи. Те ж саме відбувається і з коефіцієнтами \mathbf{A} і \mathbf{C} . Твердження доведене. [46]

Зважені політочкові перетворення. У загальному випадку мінімізуючий функціонал зваженого політочкового перетворення можна записати у вигляді:

$$S = \sum_{i=1}^p m(\beta_i, \beta_j)^2 (\omega_i - \omega_j)^2, \quad 1 \leq j \leq p.$$

Або у частинному випадку:

$$S = \sum_{i=1}^p m(\beta_i) (\omega_i - 1)^2. \quad (3.5)$$

Вагові коефіцієнти $m(\beta_i)$ можуть бути задані у числовому вигляді й вводиться до перетворення, чи мати певне функціональне подання й визначатись у процесі політочкового перетворення відповідно з конфігураціями базису образу й прообразу. [47]

В залежності від умов конкретної задачі чи вимог до результату певний вигляд функціонального завдання вагових функцій політочкового перетворення вибирається з точки зору потреб користувача. [48] Наприклад, будемо вважати, що кожна точка каркасу впливає на політочкову залежність тим сильніше, чим ближче вона знаходиться до конкретної точки об'єкта, що перетворюється. Розглянемо таку залежність докладніше на прикладі застосування вагових коефіцієнтів до функціоналу (3.5).

Нехай $m(\beta_i) = \frac{1}{\beta_i^2}$, де β_i - відстань від об'єкта перетворень (прямої) до точок

базису. Таким чином, ми вводимо умову, за якої вагові коефіцієнти точок базису політочкового перетворення обернено пропорційно залежать від квадрату відстані до зазначеної точки, тобто чим ближче точка до зазначеної прямої, тим більше пряма до неї притягується, і навпаки, далекі точки, які знаходяться на більшій відстані, мало впливають на перетворення прямої. Чим більша відстань від точки до прямої, тим менше її вплив на політочкові перетворення зазначеної прямої.

$$S = \sum_{i=1}^p \left(\left(\frac{1}{\beta_i^2} \right) (\omega_i - 1)^2 \right) \rightarrow \min.$$

Мінімізуємо заданий функціонал. Підставимо замість ω_i вираз φ_i / β_i :

$$S = \sum_{i=1}^p \left(\left(\frac{1}{\beta_i^2} \right) (\omega_i - 1)^2 \right) =$$

$$= \sum_{i=1}^p \left(\left(\frac{1}{\beta_i^2} \right) \left(\left(\frac{(\mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{y}_i + \mathbf{C}\mathbf{z}_i)}{\beta_i} \right) - 1 \right)^2 \right) \rightarrow \min.$$

Необхідно знайти \mathbf{A} , \mathbf{B} , \mathbf{C} , які є розв'язком зазначеної системи, для чого знаходимо частинні похідні по \mathbf{A} та по \mathbf{B} :

$$\frac{\partial S}{\partial \mathbf{A}} = 2 \sum_{i=1}^p \left(\left(\frac{1}{\beta_i^2} \right) \left(\left(\frac{(\mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{y}_i + \mathbf{C}\mathbf{z}_i)}{\beta_i} \right) - 1 \right) \left(\frac{\mathbf{x}_i}{\beta_i} \right) \right) = \mathbf{0};$$

$$\frac{\partial S}{\partial \mathbf{B}} = 2 \sum_{i=1}^p \left(\left(\frac{1}{\beta_i^2} \right) \left(\left(\frac{(\mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{y}_i + \mathbf{C}\mathbf{z}_i)}{\beta_i} \right) - 1 \right) \left(\frac{\mathbf{y}_i}{\beta_i} \right) \right) = \mathbf{0};$$

Розкриємо дужки :

$$\mathbf{A} \sum_{i=1}^p \left(\frac{\mathbf{x}_i \mathbf{x}_i}{\beta_i^4} \right) + \mathbf{B} \sum_{i=1}^p \left(\frac{\mathbf{x}_i \mathbf{y}_i}{\beta_i^4} \right) + \mathbf{C} \sum_{i=1}^p \left(\frac{\mathbf{x}_i \mathbf{z}_i}{\beta_i^4} \right) - \sum_{i=1}^p \frac{\mathbf{x}_i}{\beta_i^4} = \mathbf{0};$$

$$\mathbf{A} \sum_{i=1}^p \left(\frac{\mathbf{y}_i \mathbf{x}_i}{\beta_i^4} \right) + \mathbf{B} \sum_{i=1}^p \left(\frac{\mathbf{y}_i \mathbf{y}_i}{\beta_i^4} \right) + \mathbf{C} \sum_{i=1}^p \left(\frac{\mathbf{y}_i \mathbf{z}_i}{\beta_i^4} \right) - \sum_{i=1}^p \frac{\mathbf{y}_i}{\beta_i^4} = \mathbf{0};$$

У спрощеному вигляді за умови $\mathbf{z}_i = \mathbf{1}$ отримаємо таку систему:

$$\begin{cases} \mathbf{A}\mathbf{x}_{10} + \mathbf{B}\mathbf{y}_{10} + \mathbf{C} = \mathbf{d}_{10}, \\ \mathbf{A}\mathbf{x}_{20} + \mathbf{B}\mathbf{y}_{20} + \mathbf{C} = \mathbf{d}_{20}, \\ \mathbf{A}^2 + \mathbf{B}^2 = \mathbf{1}, \end{cases}$$

де

$$\mathbf{x}_{10} = \frac{\sum_{i=1}^p \mathbf{x}_i^2}{\sum_{i=1}^p \beta_i^4}, \quad \mathbf{y}_{10} = \frac{\sum_{i=1}^p \mathbf{x}_i \mathbf{y}_i}{\sum_{i=1}^p \beta_i^4}, \quad \mathbf{d}_{10} = \frac{\sum_{i=1}^p \mathbf{d}_i \mathbf{x}_i}{\sum_{i=1}^p \beta_i^4},$$

$$\mathbf{x}_{20} = \frac{\sum_{i=1}^p \mathbf{x}_i}{\sum_{i=1}^p \beta_i^4}, \quad \mathbf{y}_{20} = \frac{\sum_{i=1}^p \mathbf{y}_i}{\sum_{i=1}^p \beta_i^4}, \quad \mathbf{d}_{20} = \frac{\sum_{i=1}^p \mathbf{d}_i}{\sum_{i=1}^p \beta_i^4},$$

$$x_{20} = \frac{\frac{\sum_{i=1}^p x_i y_i}{\sum_{i=1}^p \beta_i^4}}{\frac{\sum_{i=1}^p y_i}{\sum_{i=1}^p \beta_i^4}}, \quad y_{20} = \frac{\frac{\sum_{i=1}^p y_i^2}{\sum_{i=1}^p \beta_i^4}}{\frac{\sum_{i=1}^p y_i}{\sum_{i=1}^p \beta_i^4}}, \quad d_{20} = \frac{\frac{\sum_{i=1}^p d_i y_i}{\sum_{i=1}^p \beta_i^4}}{\frac{\sum_{i=1}^p y_i}{\sum_{i=1}^p \beta_i^4}},$$

де x_i, y_i - декартови координати точок базису, а β_i – відстань від точок базису до прямої.

Ця система дає A, B, C - коефіцієнти нової прямої при зважених політочкових перетвореннях.

Політочкові перетворення на площині з використанням лінійної системи рівнянь. [77] При моделюванні динамічних процесів часто виникає проблема конструювання об'єктів (плоских чи об'ємних), що зазнали деформаційних змін. Розглянемо інший підхід до вирішення оптимізаційної задачі. [78] Для цього розглянемо можливість розв'язання задачі політочкових перетворень на площині на базі лінійної системи рівнянь.

Початковий базис в однорідному просторі задається базисними точками з координатами $x_i'', y_i'', z_i'', i=1,2,...,p$.

У цьому базисі пряма-прообраз визначається за допомогою рівнянь вигляду

$$ax_i'' + by_i'' + cz_i'' = \beta_i, \quad i = 1, ..., p,$$

де x_i'', y_i'', z_i'' - координати точок початкового базису, a, b, c – коефіцієнти заданої прямої.

Перетворений базис буде заданий у вигляді сім'ї нових точок x_i, y_i, z_i . [79] Нові “відстані” будуть визначатись за формулами:

$$\varphi_i = Ax_i + By_i + Cz_i,$$

де x_i, y_i, z_i - координати точок нового базису, A, B, C – невідомі коефіцієнти перетвореної прямої у новому базисі.

Для отримання однозначного розв'язку задачі політочкового перетворення прямої з одного базису в інший треба було застосовувати оптимізаційні методи, які зводилися до розв'язання нелінійної системи рівнянь вигляду

$$\begin{cases} \mathbf{Ax}_{10} + \mathbf{By}_{10} + \mathbf{C} = \mathbf{d}_{10}, \\ \mathbf{Ax}_{20} + \mathbf{By}_{20} + \mathbf{C} = \mathbf{d}_{20}, \\ \mathbf{A}^2 + \mathbf{B}^2 = \mathbf{1}, \end{cases}$$

де $\mathbf{x}_{10}, \mathbf{x}_{20}, \mathbf{y}_{10}, \mathbf{y}_{20}, \mathbf{d}_{10}, \mathbf{d}_{20}$ – деякі функції від декартових координат точок нового базису та від відстаней прямої прообразу до точок висхідного базису. [80] Ця система дає нам $\mathbf{A}, \mathbf{B}, \mathbf{C}$ – коефіцієнти нової прямої, які отримуються за раніше описаними формулами.

Існує інший підхід до розв'язання цієї задачі.

Нелінійність вказаній системі рівнянь надає третє рівняння, яке має на меті зберігати нормованість перетвореної прямої. [49] Нормою у випадку таких політочкових перетворень є відстань від точок базису до прямої, що перетворюється, причому під відстанню розуміють число, яке може бути отримане при підстановці координат точок базису у нормований вигляд прямої. [81] Якщо не накладати вимоги нормованості отриманої (перетвореної) прямої, то під відстанню (нормою) можна розуміти число, яке отримане шляхом прямої підстановки координат точок базису у рівняння прямої.

У цьому випадку третє рівняння системи (як і перші два) ми отримаємо при мінімізації оптимізаційного функціоналу такого вигляду:

$$S = \sum_{i=1}^p (\omega_i - 1)^2 \rightarrow \min.$$

Для цього підставимо замість ω_i вираз $\frac{\omega_i}{\beta_i}$:

$$S = \sum_{i=1}^p \left(\frac{\varphi_i}{\beta_i} - 1 \right)^2 = \sum_{i=1}^p \left(\left(\frac{(\mathbf{Ax}_i + \mathbf{By}_i + \mathbf{CZ}_i)}{\beta_i} \right) - 1 \right)^2 \rightarrow \min.$$

Необхідно знайти $\mathbf{A}, \mathbf{B}, \mathbf{C}$, які є розв'язком системи, для чого знаходимо частинні похідні за трьома змінними \mathbf{A}, \mathbf{B} та \mathbf{C} :

$$\frac{\partial S}{\partial A} = 2 \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + Cz_i)}{\beta_i} \right) - 1 \right) \left(\frac{X_i}{\beta_i} \right) = 0;$$

$$\frac{\partial S}{\partial B} = 2 \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + Cz_i)}{\beta_i} \right) - 1 \right) \left(\frac{Y_i}{\beta_i} \right) = 0;$$

$$\frac{\partial S}{\partial C} = 2 \sum_{i=1}^p \left(\left(\frac{(Ax_i + By_i + Cz_i)}{\beta_i} \right) - 1 \right) \left(\frac{Z_i}{\beta_i} \right) = 0;$$

Після того, як ми у кожному з трьох рівнянь розкриємо дужки та зведемо подібні, отримаємо три рівняння з невідомими **A**, **B**, **C** такого вигляду:

$$A \sum_{i=1}^p \frac{(X_i X_i)}{\beta_i^2} + B \sum_{i=1}^p \frac{(X_i Y_i)}{\beta_i^2} + C \sum_{i=1}^p \frac{(X_i Z_i)}{\beta_i^2} - \sum_{i=1}^p \frac{X_i}{\beta_i^2} = 0;$$

$$A \sum_{i=1}^p \frac{(Y_i X_i)}{\beta_i^2} + B \sum_{i=1}^p \frac{(Y_i Y_i)}{\beta_i^2} + C \sum_{i=1}^p \frac{(Y_i Z_i)}{\beta_i^2} - \sum_{i=1}^p \frac{Y_i}{\beta_i^2} = 0;$$

$$A \sum_{i=1}^p \frac{(Z_i X_i)}{\beta_i^2} + B \sum_{i=1}^p \frac{(Z_i Y_i)}{\beta_i^2} + C \sum_{i=1}^p \frac{(Z_i Z_i)}{\beta_i^2} - \sum_{i=1}^p \frac{Z_i}{\beta_i^2} = 0;$$

Ці три рівняння утворюють систему, яка є лінійною відносно невідомих **A**, **B**, **C** і тому може бути розв'язана будь-яким відомим методом. [50]

Розв'язком даної системи будуть **A**, **B**, **C**, які є коефіцієнтами нової перетвореної прямої.

4. ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ РОЗРОБКИ СИСТЕМИ МОДЕЛЮВАННЯ ПРОЦЕСІВ ДЕФОРМАЦІЇ ГЕОМЕТРИЧНИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ ПОЛІКООРДИНАТНИХ ПРЕРЕТВОРЕНЬ

Для розробки програмної системи було обрано такий інструментарій:

- мова програмування JavaScript;
- технологія WebGL;
- бібліотека Three.js;
- мова програмування PHP;
- фреймворк Laravel;
- база даних MS SQL Server;

4.1. Технології для розробки інтерфейсу

Для створення клієнтської частини системи були використані такі технології як HTML5, CSS3, JavaScript, а також додаткові бібліотеки JQuery та Bootstrap, які автоматизують створення інтерфейсу та спрощують доступ до різних частин порталу.

Сама структура інтерфейсу була реалізована за допомогою мови розмітки HTML, зокрема HTML5.

HTML впроваджує засоби для:

- створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;

— включення зображень, звуку, відео, та інших об'єктів до тексту.

Щодо використання самої останньої версії цієї мови, то вона має ряд суттєвих переваг, так як при прийнятті рішення про введення нових тегів було розглянуто більшість популярних сайтів і виділено основні елементи, які були спільними для всіх веб-сторінок.

Розмічаючи області на сторінці за допомогою певних елементів, ця технологія може допомогти полегшити користувачеві навігацію. Наприклад, він може легко пропустити розділ навігації або швидко переходити від однієї статті до іншої без необхідності для авторів робити відповідні посилання. Автори також отримують вигоду в результаті заміни великої кількості div-ів одним з декількох відповідних елементів, що також приводить до чистого і легкого для автора початкового коду.

Елементами header є заголовки розділів. Вони можуть складатися з декількох частин — наприклад, було б виправдано розділяти блок заголовка на підзаголовки, історію версій або вказання авторства. Елемент footer визначає нижню частину розділу, до якого він відноситься. Зазвичай він містить інформацію про розділ — наприклад, ім'я автора, посилання на схожі документи, копірайт і тому подібне. Блок nav містить список посилань для навігації. Підходить, наприклад, для навігації по сайту, або для змісту. Елемент aside підходить для розміщення вмісту яким-небудь чином спорідненого основному контенту. У звичайному випадку буде корисний для розмітки бічної колонки. Тег section представляє загальний розділ документа або додатку, наприклад, такий як розділ. Тег article відзначає незалежний розділ документа, сторінки або сайту. Застосовується для такого вмісту як новини, запису блога, повідомлення у форумі або коментарі користувачів.

Підтримкою для HTML є CSS.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай

HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на

екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.).

Переваги CSS:

- інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;

- різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;

- сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;

- прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

Взаємодія користувача з інтерфейсом відбувається за допомогою мови програмування JavaScript.

JavaScript (JS) — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером,

асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні

архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Проблему реалізації текстового візуального редактора було досягнуто за допомогою технології bootstrap-wysiwyg, фреймворка Bootstrap, що розроблений на основі CSS та JavaScript.

Bootstrap — це безкоштовний набір інструментів з відкритим сирцевим кодом, призначений для створення веб-сайтів та веб-застосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript.

Основні інструменти Bootstrap:

сітки (grid) — наперед задані, готові до використання колонки;

шаблони (template) — фіксовані чи адаптивні шаблони сторінок;

типографіка (typography) — опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо;

мультимедіа (media) — засоби управління зображеннями та відео;

таблиці (table) — засоби оформлення таблиць, які зокрема забезпечують сортування;

форми (form) — класи для оформлення як форм, так і деяких подій;

навігація (nav, navbar) — класи для оформлення вкладок, сторінок, меню і панелей навігації;

сповіщення (alert) — класи для оформлення діалогових вікон, підказок і спливаючих вікон;

іконочний шрифт (icon font) — набір іконок у вигляді шрифту, складається майже з 500 компонентів.

Текст редагується в div з атрибутом contenteditable, без створення iframe і textarea. У ньому немає стандартної панелі інструментів та стилів — їх можна створювати засобами Bootstrap. Команди редагування прив'язуються до клавіш за допомогою атрибуту data-edit.

jQuery - бібліотека JavaScript, що містить в собі готові функції мови JavaScript, всі операції jQuery виконуються з коду JavaScript. Бібліотека jQuery проводить маніпуляції з html-елементами, керуючи їх поведінкою і використовуючи DOM для зміни структури веб-сторінки. При цьому вихідні файли HTML і CSS не змінюються, зміни вносяться лише в відображення сторінки для користувача. Для вибору елементів використовуються селектори CSS. Вибір здійснюється за допомогою функції \$(). При виклику функція \$() повертає новий екземпляр об'єкта JQuery, який обертає нуль або більше елементів DOM і дозволяє взаємодіяти з ними різними способами. Виконання різних сценаріїв можливо тільки після закінчення завантаження структури документа document, коли браузер перетворює html-код сторінки в дерево DOM. Можливості:

- Ядро крос-браузерних CSS-селекторів;
- Перехід по дереву DOM, включаючи підтримку XPath як плагіна;
- Трігери подій;
- Візуальні ефекти;
- AJAX-додатки
- JavaScript-плагіни;

4.2. Інструменти для роботи з графікою. WebGL і Three.js

Для організації роботи з графікою було виористано технологію WebGL. WebGL являє собою технологію, що базується на OpenGL ES 2.0 і призначену для малювання і відображення інтерактивної 2D- і 3D-графіки в веб-браузерах. При цьому для роботи з даною технологією не потрібні сторонні плагіни або бібліотеки. Вся робота веб-додатків з використанням WebGL заснована на коді JavaScript, а деякі елементи коду - шейдери можуть виконуватися безпосередньо на графічних

процесорах на відкритих, завдяки чому розробники можуть отримати доступ до додаткових ресурсів комп'ютера, збільшити швидкодію. Таким чином, для створення додатків розробники можуть використовувати стандартні для веб-середовища технології HTML / CSS / JavaScript і при цьому також застосовувати апаратне прискорення графіки. Якщо створення настільних додатків працюють з 2d і 3d-графікою нерідко обмежується цільовою платформою, то тут головним обмеженням є тільки підтримка браузером технології WebGL. А самі веб-додатки, побудовані з використанням даної платформи, будуть доступні в будь-якій точці земної кулі при наявності мережі інтернет незалежно від використовуваної платформи: чи то це десктопи з ОС Windows, Linux, Mac, то це смартфони та планшети, то чи це ігрові консолі. WebGL виник з експериментів над Canvas 3D американського розробника сербського походження Володимира Вукічевіч з компанії Mozilla в 2006 році. Згодом розробники браузерів Opera і Mozilla стали створювати свої реалізації WebGL. А згодом була організована робоча група за участю найбільших розробників браузерів Apple, Google, Mozilla, Opera для роботи над специфікацією технології. І в 3 березня 2011 року було представлена специфікація WebGL 1.0. Перевагами використання WebGL є:

- Кросбраузерність і відсутність прив'язки до певної платформи. Windows, MacOS, Linux, iOS, Android - все це не важливо, головне, щоб ваш браузер підтримував WebGL;
- Використання мови JavaScript, який досить поширений;
- Автоматичне управління пам'яттю. На відміну від OpenGL в WebGL не треба виконувати спеціальні дії для виділення і очищення пам'яті;
- Оскільки WebGL для рендеринга графіки використовує графічний процесор на відеокарті (GPU), то для цієї технології характерна висока продуктивність, яка порівнянна з продуктивністю нативних додатків.

Для оптимізації коду та прискорення розробки було використано бібліотеку Three.js, що базується на технології WebGL. Three.js - це кросбраузерна бібліотека JavaScript, що використовується для створення та відображення анімованої комп'ютерної 3D графіки при розробці веб-додатків. Three.js скрипти можуть

використовуватися спільно з елементом HTML5 CANVAS, SVG або WebGL. Вихідний код розташований в репозиторії GitHub. Three.js дозволяє створювати прискорену на GPU 3D графіку, використовуючи мову JavaScript, як частина сайту без підключення пропрієтарних плагінів для браузера. Це можливо завдяки використанню технології WebGL. Підтримує тривимірні моделі формату Collada. Особливостями Three.js є:

Рендерери: Canvas, SVG або WebGL;

Сцена: додавання і видалення об'єктів в режимі реального часу;

Камери: перспективна або Ортографічна;

Анімація: каркаси, пряма кінематика, інверсна кінематика, покадрова анімація;

Джерела світла: зовнішній, спрямований, точковий; тіні: кинуті і отримані;

Шейдери: повний доступ до всіх OpenGL-шейдерам (GLSL);

Об'єкти: сітки, частинки, спрайт, лінії, скелетна анімація і інше;

Геометрія: площину, куб, сфера, тор, 3D текст і інше; модифікатори: тканина, видавлювання;

Завантажники даних: двійковий, зображення, JSON і сцена;

Експорт та імпорт: утиліти, що створюють Three.js-сумісні JSON файли з форматів: Blender, openCTM, FBX, 3D Studio Max, і Wavefront.obj файл

Підтримка: документація по API бібліотеки знаходиться в процесі постійного розширення і доповнення, є публічний форум і велике співтовариство;

Бібліотека Three.js працює у всіх браузерах, які підтримують технологію WebGL; також може працювати з «чистим» інтерфейсом елемента CANVAS, завдяки чому працює і на багатьох мобільних пристроях. Three.js поширюється під ліцензією MIT license.

4.3. Мова програмування PHP. Фреймворк Laravel

В якості мови програмування був обраний PHP. Ця мова програмування була обрана через те, що це найпоширеніша мова для роботи з вмістом web-сторінок. За допомогою PHP можна працювати з усіма аспектами створення веб-ресурсу. Давайте розглянемо цю мову програмування трохи детальніше.

Команда розробників PHP включає в себе десятки розробників, а також десятки інших організацій, що працюють над пов'язаними з PHP і його підтримкою проектами, такими як PEAR, PECL і документацією, а також базову інфраструктуру мережі більш ніж зі ста серверів на шести з семи континентах світу. Грунтуючись на статистиці минулих років, можна з упевненістю припустити, що PHP тепер встановлений на десятки або навіть, можливо, сотні мільйонів доменів по всьому світу.

Далі ми можемо переглянути, які саме переваги надає нам використання мови програмування PHP.

Перевагою PHP є гнучкість у використанні. PHP має ресурси для виконання будь-яких задач, які можуть бути перед програмістом. PHP має 5 важливих та характеризуючих його характеристик:

- традиційність;
- простота;
- ефективність;
- безпека;
- гнучкість.

Ще одна досить важлива характеристика PHP – це безкоштовне використання цієї мови. Також початкові коди знаходяться у вільному доступі.

Це традиційна мова програмування, по причині того, що PHP будувався на основі таких мов програмування, як C і Perl. Багато програмістів можуть швидко її освоїти, не витрачаючи багато часу.

Синтаксис PHP подібний синтаксису мови Cі. Деякі елементи, такі як асоціативні масиви і цикл `foreach`, запозичені з Perl.

Для роботи програми не потрібно описувати будь-які змінні, використовувати модулі та інше. Будь-яка програма може починатися безпосередньо з оператора PHP.

Відкриваючий тег виду `<? =` використовується для скороченого запису конструкцій використовуваних для виведення рядка.

PHP виконує код, що знаходиться всередині обмежувачів, таких як `<? Php?>`. Все, що знаходиться поза обмежувачів, виводиться без змін.

Крім обмежувачів `<? Php?>`, допускається використання скороченого варіанту `<? ?>`. Крім того, до версії 7.0 допускалося використання обмежувачів мови програмування ASP `<% %>` і `<script language = "php"> </ script>`. Робота скорочених конструкцій визначається в конфігураційному файлі `php.ini`.

Імена змінних починаються з символу `$`, тип змінної оголошувати не потрібно. Імена змінних і констант чутливі до регістру символів. Імена класів, методів класів і функцій до регістру символів не чутливі. Змінні обробляються в рядках, ув'язнених в подвійні лапки, і heredoc-рядках (рядках, створених за допомогою оператора `<<<`). Змінні в рядках, ув'язнених в одинарні лапки, не обробляються.

PHP розглядає перехід на новий рядок як пробіл, так само як HTML і інші мови з вільним форматом. Інструкції поділяються за допомогою крапки з комою (`;`), за винятком деяких випадків, після оголошення конструкції `if / else` і циклів.

Змінні в функцію можна передавати як за значенням, так і за посиланням (використовується знак `&`).

PHP підтримує три типи коментарів: в стилі мови Cі (обмежені `/ * * /`), C ++ (починаються з `//` і йдуть до кінця рядка) і оболочки UNIX (з `#` до кінця рядка).

PHP є мовою програмування з динамічною типізацією, що не вимагає вказівки типу при оголошенні змінних, так само як і самого оголошення змінних. Перетворення між скалярними типами часто здійснюються неявно без додаткових зусиль (втім, PHP надає широкі можливості і для явного перетворення типів).

Скалярними типами даних є:

цілочисельний тип (integer);
число з плаваючою точкою (float, double);
логічний тип (boolean);
строковий тип (string).

Нескалярними типами даних є:

масив (array);
об'єкт (object);
зовнішній ресурс (resource);
невизначене значення (null).

Псевдотипами даних є:

mixed будь-який тип;
number число (integer або float);
callback (string або анонімна функція);
void відсутність параметрів.

Діапазон цілих чисел (integer) в PHP залежить від платформи (зазвичай, це діапазон 32-бітних знакових цілих чисел, тобто, від -2 147 483 648 до 2 147 483 647). Числа можна задавати в десятковій, вісімковій і шістнадцятковій системах числення. Діапазон дійсних чисел (double) також залежить від платформи (для 32-бітної архітектури діапазон дозволяє оперувати числами від $\pm 1.7 \times 10^{-308}$ до $\pm 1.7 \times 10^{+308}$).

PHP надає розробникам логічний тип (boolean), здатний приймати тільки два значення TRUE («істина») і FALSE («брехня»). При перетворенні в логічний тип число 0, порожній рядок, нуль в рядку «0», NULL і порожній масив вважаються рівними FALSE. Всі інші символи будуть автоматично перетворюються в TRUE.

Спеціальний тип null призначений для змінних без певного значення. Єдиним значенням даного типу є константа NULL. Тип null приймають неініціалізовані змінні, змінні початкові константою NULL, а також змінні, вилучені за допомогою конструкції unset ().

Посилання на зовнішні ресурси мають тип «ресурс» (resource). Змінні даного типу, як правило, представляють собою дескриптор, що дозволяє управляти зовнішніми об'єктами, такими як файли, динамічні зображення, результуючі таблиці бази даних та інше.

Масиви (array) підтримують числові і рядкові ключі і є гетерогенними. Масиви можуть містити значення будь-яких типів, включаючи інші масиви. Порядок елементів і їх ключів зберігається. Не зовсім коректно називати php-масиви масивами, насправді це, швидше за все, упорядкований хеш. Можливо несподіване поведінку при використанні циклу `for` з лічильником замість `foreach`. Так, наприклад, при сортуванні масиву з чисельними індексами функціями зі стандартної бібліотеки, упорядковано і ключі теж.

Покажчик на функцію в PHP може бути представлений замиканням або псевдотіпов `callback`. Замикання доступне з версії 5.3 і в коді виглядає як просте визначення функції, в яку можна явно прокинути значення з контексту.

Callback тип може бути представлений:

рядком (інтерпретується як назва функції);

масивом де нульовий і перший елемент - рядки (інтерпретується як назва статичної функції класу);

масивом де нульовий елемент - об'єкт, а перший - рядок (інтерпретується як метод у об'єкта).

Звернення до змінних здійснюється за допомогою символу `$`, за яким слідує ім'я змінної. Дана конструкція може бути застосована також для створення динамічних змінних і функцій.

Суперглобальних масивами (англ. Superglobal arrays) в PHP називаються зумовлені масиви, що мають глобальну область видимості без використання директиви `global`. Велика частина цих масивів містить вхідні дані запиту користувача (параметри GET-запиту, поля форм при посилці методом POST, куки та інших засобах).

Laravel – це безкоштовний веб-фреймворк з відкритим кодом, призначений для розробки з використанням архітектурної моделі MVC (англ. Model View

Controller - модель-уявлення-контролер). Laravel випущений під ліцензією MIT. Ключові особливості, що лежать в основі архітектури Laravel:

Пакети (англ. Packages) - дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткові можливості вже доступні у вигляді таких модулів;

Eloquent ORM - реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об'єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent;

Логіка програми - частина розробляється, оголошена або за допомогою контролерів, або маршрутів (функцій-замикань). Синтаксис оголошень схожий на синтаксис, який використовується в каркасі Sinatra;

Зворотня маршрутизація пов'язує між собою генеруємі додатком посилання і маршрути, дозволяючи змінювати останні з автоматичним оновленням пов'язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL;

REST-контролери - додатковий шар для поділу логіки обробки GET- і POST-запитів HTTP;

Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключати файли їх визначень в include. Завантаження на вимогу запобігає завантаження непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються;

Укладачі уявлень (англ. View composers) - блоки коду, які виконуються при генерації уявлення (шаблону).

Інверсія управління (англ. Inversion of Control) - дозволяє отримувати екземпляри об'єктів за принципом зворотного управління. Також може використовуватися для створення і отримання об'єктів-одинаків (англ. Singleton);

Міграції - система управління версіями для баз даних. Дозволяє зв'язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми;

Модульне тестування (юніт-тести) - грає дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання регресій (помилкам внаслідок поновлення коду або виправлення інших помилок);

Сторінковий висновок (англ. Pagination) - спрощує генерацію сторінок, замінюючи різні способи вирішення цього завдання єдиним механізмом, вбудованим в Laravel;

4.4. Середовище Sublime Text

Для розробки веб-ресурсу був обраний редактор, який задовольняє всі вимоги сучасної розробки та має багато зручних властивостей. Він простий у використанні, підтримує синтаксис мови PHP та інтуїтивно зрозумілий інтерфейс.

“Sublime Text - багатоплатформовий пропріетарний текстовий редактор. Підтримує плагіни на мові програмування Python.

Розробник дозволяє безкоштовно і без обмежень ознайомитися з продуктом, проте програма повідомляє про необхідність придбання ліцензії.

Даний текстовий редактор володіє такими властивостями:

- швидка навігація;
- командна палітра;
- API плагінів на Python;
- одначасне редагування;
- висока степнь налаштування.

Sublime Text підтримує велику кількість мов програмування і має можливість підсвічування синтаксису для C, C ++, C #, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL і XML.

На додаток до тих мов програмування, які включені за замовчуванням, користувачі мають можливість завантажувати плагіни для підтримки інших мов.

Sublime Text може бути оснащений менеджером пакетів, який дозволяє користувачеві знаходити, встановлювати, оновлювати і видаляти пакети без перезавантаження програми. Менеджер підтримує встановлені пакети в актуальному стані, завантажуючи нові версії з репозиторіїв. Крім того, він надає команди для активації і деактивації встановлених пакетів.

Редактор містить різні візуальні теми, з можливістю завантаження додаткових.

Користувач може переглянути код та список папок під виглядом мінікарти на багато рівнів. Це зручно для навігації (рисунок 4.1).

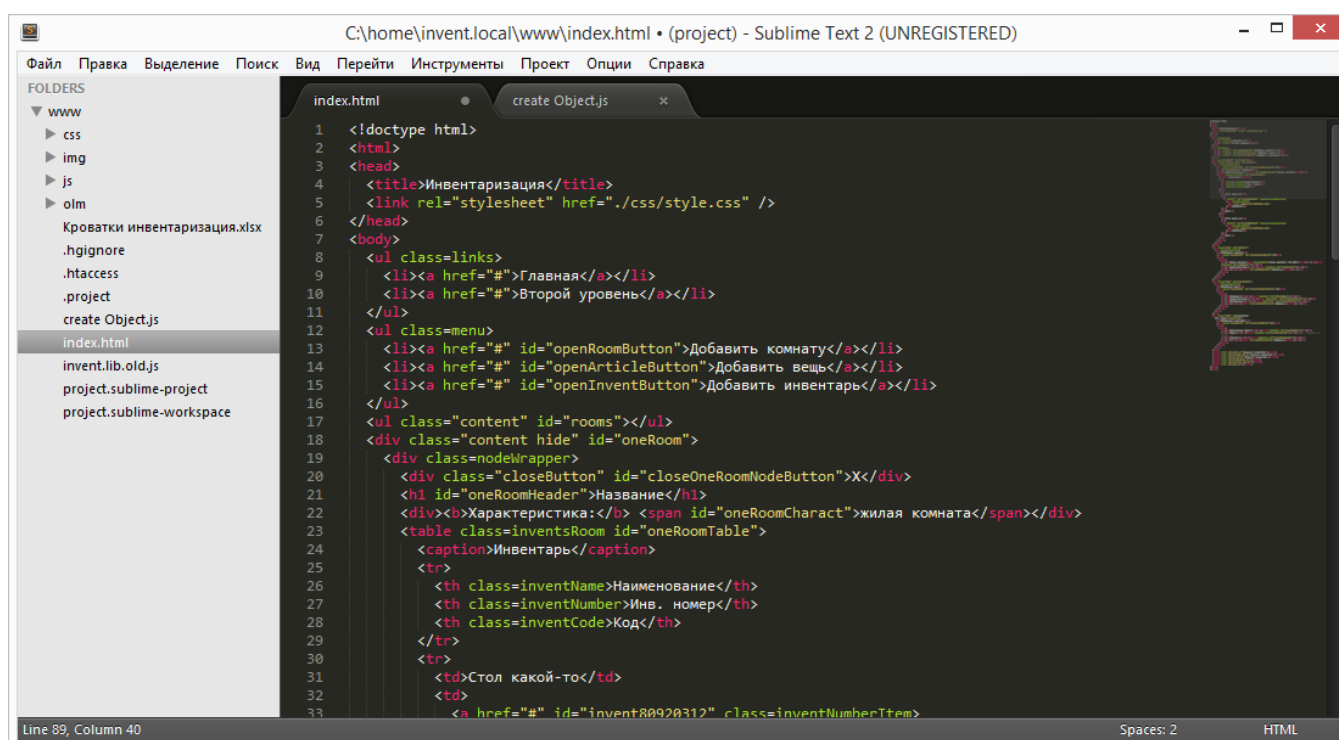


Рисунок 4.1 – Інтерфейс редактору та мінікарта Sublime Text

Доступно декілько режимів зображення. Перше має від онієї до чотирьох панелей, тому одночасно ми можемо користуватись чотирьма файлами. Повний режим дозволяє нам переглядати один файл, але на повний екран.

Користувач має можливість виділити повний стовпець, розставить декілька покажчиків по тесту. Це надає можливість користувачу робити правки швидко. Покажчики мають форму поведінки, яка відповідає тому, що кожен покажчик єдиний. Такі типи команд, як: переміщення на одне слово або його частини

(CamelCase, розділений дефісом або підкресленням), вибірка тексту, переміщення на один рядок, переміщення на один знак, перехід на початок / кінець рядка та інше, має вплив на всі покажчики, що дозволяє правити текст швидко.

При наборі користувачем тексту в Sublime Text, дивлячись яка мова використовується, програма пропонує відомі та популярні версії запису. Це правило також діє на раніше створені змінні.

Контраст Sublime Text Досягнутий завдяки чорному фону, тоді як основні елементи синтаксису мають різнокольорове забарвлення, тому їх краще видно саме на темному фоні інтерфейсу Sublime Text.

Використовуючи Sublime Text За допомогою Sublime Text є можливість запускати програми без використання командного рядка. Є можливість самостійно налаштовувати систему автоматичних збірок програми в той момент, коли користувач хоче зберегти код.

Інструмент для навігації, що надає можливість юзеру змінювати файли одним натиском миші.

Додаткові характеристики:

- можливість самостійно додавати різні комбінації клавіш для навігації та інших операцій;

- автозбереження додано як додаткова функція, що часто може бути досить зручним рішенням, так як ми не втратимо інформацію;

- функція перевірки синтаксису працює подібним же чином, перевіряючи коректність прямо під час введення;

- додавання та налаштування макросів для автоматизації;

- команди редагування, включаючи редагування відступів, переформатування параграфів і об'єднання рядків;

- для пошуку в документі також реалізовано автозаповнення під час набору запиту.

Проект почався в листопаді 2007 року з метою створити «кращий текстовий редактор», перша версія програми вийшла 18 січня наступного і була доступна тільки під Windows.

З можливостей програми варто відзначити:

- систему проектів: можливість об'єднання файлів в один проект;
- режим Distraction Free - повноекранний режим з акцентом на вмісті;
- панель швидкого доступу для переходу по файлах проекту або файлів в поточній директорії;
- можливість вибрати кілька рядків для редагування;
- API плагінів на Python;
- підтримка «гарячих» клавіш;
- пошук по файлах.

Остання версія Sublime Text 1 вийшла 13 вересня 2010 року.”

4.5. Модуль phpMyAdmin

При Вибір модулю phpMyAdmin був обумовлений його популярністю та зручністю у використанні.

Програма phpMyAdmin є одним з найпоширеніших засобів для адміністрування, управління та щоденної роботи з базами даних MySQL. Практично на кожному хостингу, де встановлена підтримка цієї СУБД, як панелі управління пропонують phpMyAdmin.

PhpMyAdmin - це набір скриптів, написаних на PHP, який надає практично всі необхідні функції по роботі з базами даних MySQL. На додаток до можливостей самого сервера MySQL, він надає додаткові фічі, які дозволяють більш ефективно і легко працювати з даними.

Адміністратори можуть легко створювати акаунти для користувачів, призначати їм привілеї, створювати і налаштовувати бази даних. Причому і користувачі, і адміністратори можуть використовувати один і той же скрипт - просто у користувача, що не володіє необхідним рівнем доступу, не буде деяких функцій - наприклад, вибору сервера БД, доступу до службових баз і баз інших

користувачів і т.д. Але вся функціональність в рамках створеної для нього бази повністю доступна.

Розробники отримують повний контроль над своїм сервером, над усіма базами і таблицями, можливість інтерактивно виконувати SQL-запити і навіть початкові функції налагодження невірних запитів. Звичайно, до відладчика з пакета MySQLQueryBrowser phpMyAdmin ще дуже далеко, але це ж зовсім різні вагові категорії. PhpMyAdmin має вбудовані засоби для проведення поточних робіт з базами і таблицями - обслуговування, перевірка і лагодження зіпсованих таблиць, резервування та відновлення з архівів, експорт даних з таблиць, як в форматі SQL, так і в більш специфічних, наприклад LaTeX або PDF. Для зменшення обсягу файл з даними можна відразу ж на сервері стиснути архіватором - це дуже корисно для створення на своїй машині копії з віддаленої бази на сайті.

Інтерфейс програми складається з двох вертикальних фреймів, в першому з них перераховані доступні бази (якщо ви адміністратор), або ж відразу таблиці вашої поточної бази даних, якщо ви не маєте доступу до інших БД (рисунок 4.2). Вибравши необхідну таблицю, ви можете перейти в вікно перегляду її параметрів і налаштувань. Сторінка завантажується в основний фрейм, так що вся інформація завжди знаходиться перед очима в одному вікні браузера.

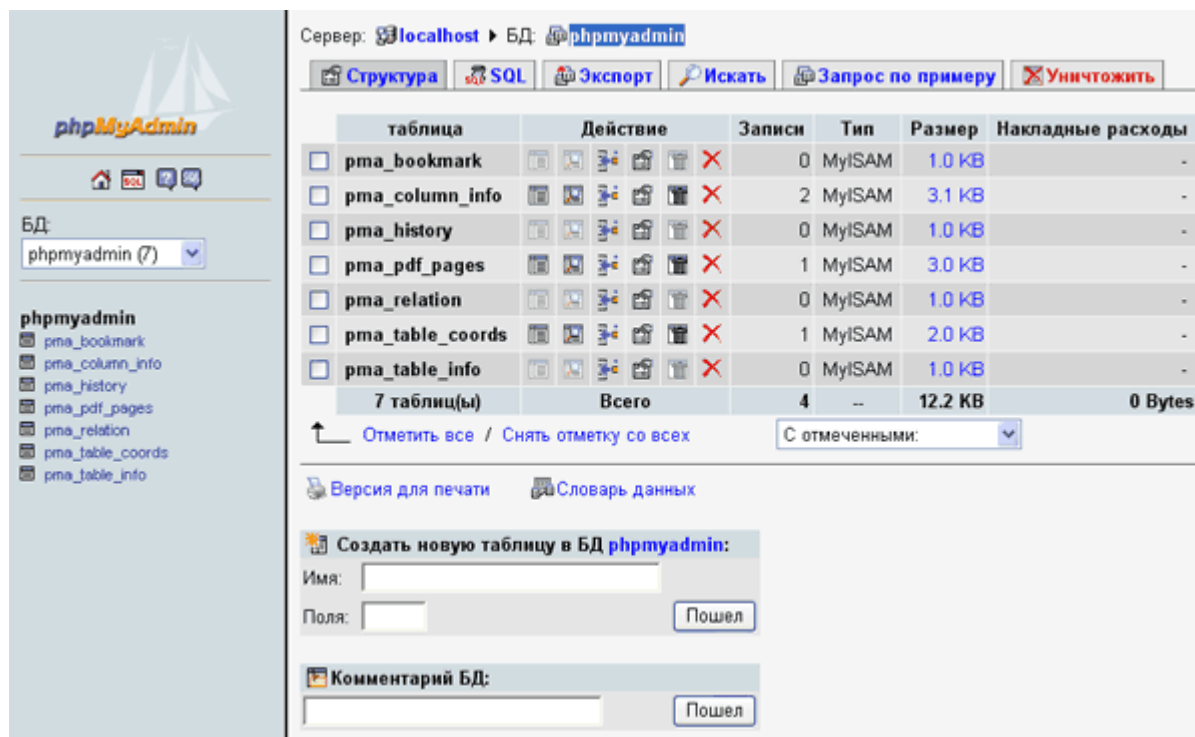


Рисунок 4.2 – Интерфейс phpMyAdmin

В основному вікні присутній ряд вкладок у верхній частині сторінки, для переходу між основними операціями над таблицею, інші ж функції доступні через посилання, які тепер найчастіше позначаються піктограмами, що дозволило отримати дуже компактний інтерфейс.

Програма дозволяє виконувати будь-які SQL-запити над базами, причому це можна робити як зі спеціального, окремого, вікна запитів, так і для кожної таблиці окремо (рисунок 4.2). На найбільш часті запити можна робити спеціальні закладки і надалі отримати доступ до них за одним натисканням на посилання.

The screenshot displays the 'Обслуживание таблицы' (Table Maintenance) section in phpMyAdmin. The main area contains several actions for the table 'pma_table_info' in the 'mysql' database:

- Изменить порядок таблицы:** A dropdown for 'db_name' (set to 'mysql'), a text input for '(отдельно)', a dropdown for 'По возрастанию', and a 'Пошел' button.
- Переименовать таблицу в:** A text input containing 'pma_table_info' and a 'Пошел' button.
- Переместить таблицы в (база данных.таблица):** A dropdown for 'mysql', a text input for 'pma_table_info', and a 'Пошел' button.
- Скопировать таблицу в (база данных.таблица):** A dropdown for 'phpmyadmin', a text input, and a 'Пошел' button. Below this are radio buttons for 'Только структуру' (selected), 'Структура и данные', and 'Только данные'. There are also checkboxes for 'Добавить удаление таблицы', 'Добавить значение AUTO_INCREMENT', and 'Переключиться на скопированную таблицу'.
- Комментарий к таблице :** A text input with 'Table information for phpMyAdmin' and a 'Пошел' button.
- Тип таблицы:** A dropdown set to 'MyISAM' and a 'Пошел' button.
- Опции таблицы:** Checkboxes for 'pack_keys', 'checksum', and 'delay key write'.

On the right side, the 'Обслуживание таблицы' section lists five actions with question mark icons:

- Проверить таблицу
- Анализ таблицы
- Починить таблицу
- Оптимизировать таблицу
- Сбросить кэш таблицы ("FLUSH")

Рисунок 4.3 – Обслуговування таблиці в phpMyAdmin

Дуже цікава можливість - після виконання запиту програма показує не тільки детальну інформацію про нього (час обробки, кількість порушених рядів), а й пропонує показати розширену інформацію про використаних даних при побудові запиту (індексах та інше, що може бути корисно при побудові складних запитів, налагодженні і ручної оптимізації). Ще одна функція дозволяє по введеному запиту автоматично будувати код на PHP. Хоча результуючий код дуже простий - всього лише змінна \$ sql, яка містить код запиту, оптимізований під синтаксис PHP, але це дуже корисна функція, особливо якщо ви спочатку налагоджувати складний запит

через інтерфейс phpMyAdmin, а потім, переконавшись у коректній видачі результатів, хочете перенести запит в свій скрипт.

Хоча більшість запитів треба прямо вводити у вигляді коректних SQL-виразів, але для операції пошуку даних існує спеціальний зручний інтерфейс (рисунок 4.4). Для цього є спеціальна вкладка «Шукати», перехід на яку відкриває сторінку, де поля для пошуку представлені у вигляді форми, заповнюючи яку можна, не особливо вдаючись у тонкощі синтаксису конкретних виразів, створити досить таки складний запит (мається на увазі, запит SELECT). Допустимі поля, варіанти порівняння та сортування результатів - все можна задати через списки, які випадають і меню, а додаткові умови пошуку можна вказати в спеціальних текстових полях. Хоча такий будівник виразів зовсім не скасовує знання особливостей роботи БД і SQL, він істотно допомагає, коли треба швидко побудувати специфічний пошуковий запит. Одна з найпростіших і до того ж зручних - SQL SELECT. При знайомстві з нею будь-який програміст зможе без проблем розповісти про те, що SQL - це мова програмування, що допомагають структурувати всі необхідні запити, а SELECT - це команда, яка використовується в комбінації з SQL. Безумовно, в поєднанні SQL SELECT дозволяють проводити всі необхідні дії, Викладені на прикладах даної сторінки.. Тому, встановив програму, яка використовується SQL SELECT, ви заощадите час своїх співробітників, а також гроші компанії.

За допомогою запиту SQL SELECT можна виконувати вибірку даних з таблиці. Запити SQL SELECT використовуються в таких SQL СУБД як MySQL, Oracle, Access і інших.

Выбрать поля (минимум одно):

id
tyle_doc
theme_doc
type_doc
langv_doc
body_doc
date_inst

☐ DISTINCT

записей на страницу :

Порядок просмотра:

☒ По возрастанию
☐ По убыванию

Добавить условия поиска (тело для условия "where"): ?

Или Выполнить "запрос по примеру" (символ подстановки: "%")

Поле	Тип	Оператор	Значение
id	bigint(20)	= <input type="text"/>	<input type="text"/>
tyle_doc	varchar(255)	LIKE <input type="text"/>	<input type="text"/>
theme_doc	tinyint(3)	= <input type="text"/>	<input type="text"/>
type_doc	tinyint(3)	= <input type="text"/>	<input type="text"/>
langv_doc	tinyint(3)	= <input type="text"/>	<input type="text"/>
body_doc	longtext	LIKE <input type="text"/>	<input type="text"/>
date_inst	datetime	= <input type="text"/>	<input type="text"/>

Рисунок 4.4 – Операція пошуку по таблицям

5. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

За допомогою наведених в 3 розділі математичних засобів було розроблено систему моделювання процесів деформації геометричних об'єктів основі полікоординатних перетворень. Функціональна схема системи наведена на рисунку 5.1.

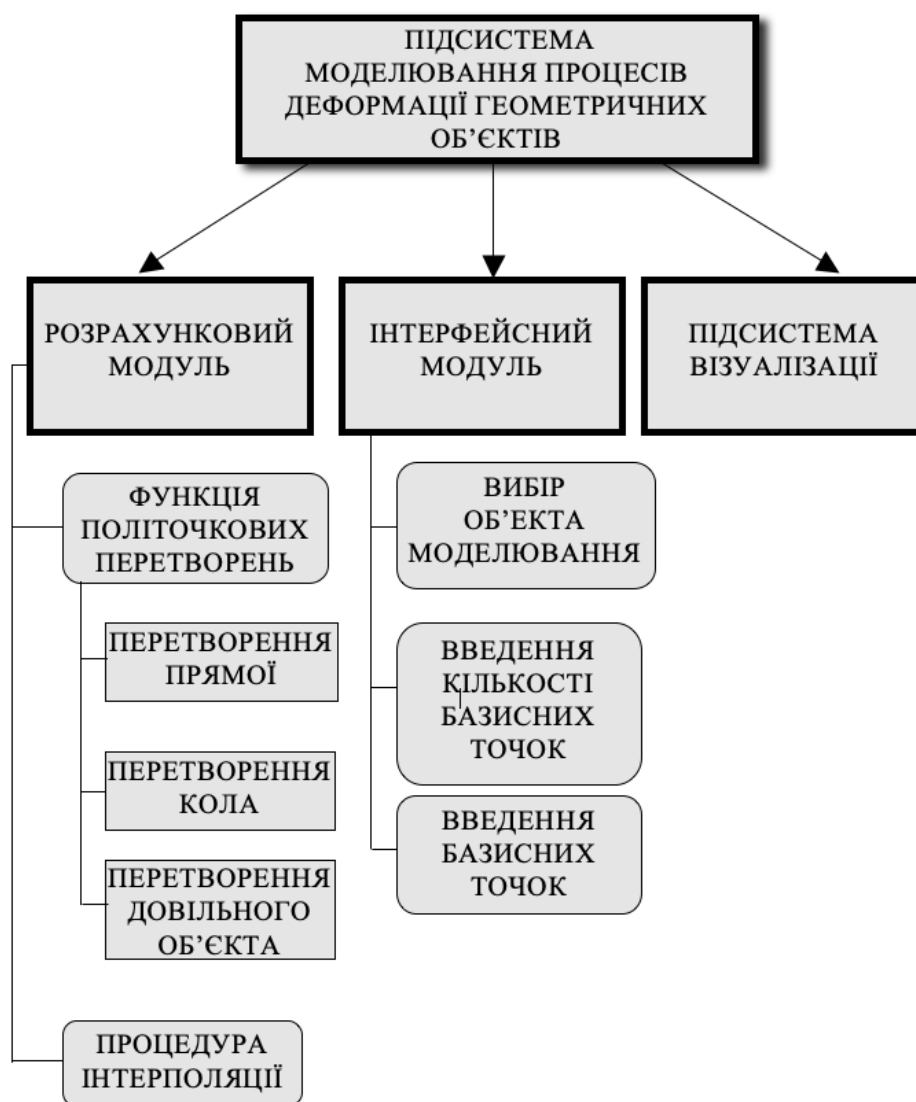


Рисунок 5.1 — Функціональна схема системи

Розрахунковий модуль системи містить функції обробки вхідних даних, а саме функції інтерполяції, політочкові перетворення та відношення площ областей, створених за допомогою політочкових перетворень та функції Гауса. [54]

Інтерфейсний модуль містить засоби керування вхідними даними.

Модуль візуалізації містить функції для відображення на екрані вхідних даних, результатів роботи програми та додаткової інформації.

5.1. Розрахунковий модуль

Одна з функцій, що міститься в розрахунковому модулі — визначення потенційних збитків від лісової пожежі. Докладний алгоритм проведення обрахунків було наведено у 3 розділі.

Для реалізації алгоритму знадобились деякі з функцій бібліотеки Math, що є частиною стандартної директиви System, яка підключається до проекту умовною конструкцією using System.

Для визначення збитків користувач повинен ввести деякі з параметрів, які не можуть бути запрограмовані заздалегідь. Ці параметри вводяться в текстове поле, яке повертає змінну типу string. Змінну такого типу неможливо використовувати для математичних обрахунків, тому виникла необхідність реалізувати функцію Convert, що конвертує деякі типи в інші. В нашому випадку була використана функція Convert.ToDouble(змінна), щоб на виході отримати змінну числового типу з плаваючою комою для більш точного результату розрахунків.

Також в розрахунковому модулі міститься реалізація політочкових перетворень, докладніше про яку було написано в розділі 3.

Для розрахунку за наданими формулами програма використовує перелік математичних функцій: pow() і exp(), sin(), cos().

Найзручнішим методом розв'язання задачі інтерполяції певним методом є написання функції, що вертає значення інтерполяційної функції у заданій точці. Тому всі функції, що відповідають безпосередньо за метод інтерполяції, організовані саме таким чином. Для цього функції були оголошені як такі, що

повертають значення. Для цього необхідно перед назвою функції при її оголошенні вказати тип значення, що повертається. Всередині опису функції за допомогою команди `return` значення можна вказати значення якої саме змінної буде повернене в результаті роботи функції. В цій програмі усі функції розрахунку інтерполяції повертають шукане значення функції для одної заданої точки.

5.2. Інтерфейсний модуль

Інтерфейсний модуль відповідає передусім за забезпечення механізмів введення інформації для обробки програмою. Розглянемо передусім функції управління вхідними даними.

У цій роботі в більшості випадків користувач вводить дані у поля для вводу. Усі поля для вводу сприймають введені значення як текст. Отже постає необхідність отримати числову інтерпретацію введених значень, оскільки користувач оперує числами, що характеризують точки або параметри функцій. За перетворення типів даних відповідає операція `Convert`, що вже була згадана у попередньому підрозділі.

Для зручності роботи з програмою користувачу пропонується вводити координати опорних точок геометричного об'єкта та точок базисів через взаємодію з робочою зоною системи, що є інтуїтивно зрозумілим. Введені таким чином координати напряду використовуються для обрахунків без необхідності їх форматувати.

Нові отримані координати обробляються класом `Converter`, що перетворює їх на плоскі координати, що можуть бути використані для побудови полігону ураженої території.

Для реалізації появи/зникнення елементів з вікна відповідає властивість елементів `Visible`. Ця властивість має тип `bool` і приймає значення `true` або `false`. При значенні `true` елемент видимий в процесі виконання програми, інакше його можна побачити лише у вікні редагування та налагодження програми. Результати обчислень виводяться в поле `label`, яке до отримання цих самих результатів не відображується звичайному користувачу.

При введенні координат користувач має можливість додавати нові координати, змінювати вже існуючі або видаляти непотрібні. Результат таких маніпуляцій буде одразу відображено на карті після натискання відповідної клавіші.

5.3. Модуль візуалізації

Основним елементом візуального представлення даних у програмі є елемент типу WebGL на головній формі, що представляє собою планшет для зображення на ньому даних.

Додавання точок відбувається шляхом натискання лівої клавіші миші по робочій зоні. Після цього обрана точка буде автоматично додана до набору координат.

Для графічного відображення маркерів на полі було використано стандартну функції Three.js, таку як `add()` (`gmap, new Point(Convert.ToDouble(point.Key), Convert.ToDouble(point.Value))`). Перший параметр цієї функції відповідає за саму мапу, на якій буде додано маркер, другий параметр містить точку, в якій цей маркер необхідно встановити.

5.4. Модульна система

“Модульне програмування набуло широкого поширення. Під модулізацією розуміється поділ програми на частини за деякими встановленими правилами. Цими частинами можуть бути програмні секції (в Кобола), внутрішні процедури (Паскаль) або зовнішні процедури (в фортрані, Кобол). У процедурних (алгоритмічних) системах програмування є спеціальна бібліотека програм (модулів) та засоби роботи з цією бібліотекою. В міру ускладнення функцій, покладених на [62] ЕОМ, росла і складність програм. Спочатку бібліотека програм використовувалася для зменшення дублювання в роботі програміста. Потім при створенні складних ПС можливості

процедурних систем програмування (мається на увазі можливість організації бібліотеки модулів) почали використовуватися для забезпечення створення складних програм одним або колективом програмістів. [82] Це пояснювалося тим, що складність прикладних програм зростає настільки, що важко було писати її як єдине ціле або розробляти одним програмістом. Програми стали занадто великі, щоб їх можна було уявити у всіх подробицях як єдине ціле і помістити в пам'яті ЕОМ цілком.

Для розміщення великих програм в пам'яті машини стали застосовуватися структури з перекриттями (оверлейні структури). Для цього було потрібно розбиття програм на модулі. Пішли шляхом подальшого розвитку і вдосконалення процедурного програмування і бібліотек програм. З появою ЕОМ з сторінковою (віртуальною) пам'яттю, управління оверлейними структурами здійснюється автоматично, але ефективність розподілу програми по сторінках залежить від обраного програмістом способу поділу її на модулі.

Програми розбиваються по модулях по причині:

- понизити види до складності розробки і реалізації;
- полегшити читання програми;
- спростити їх налаштування та внесення змін;
- робота з даними зі складною структурою спрощується;
- надмірна деталізація алгоритмів більше не буде проблемою;
- краще використання пам'яті ЕОМ при встановленні програм.

Основна риса модульного програмування - це стандартизація і паспортизація інтерфейсу між окремими програмними одиницями. [61] Модуль - це окремо функціонально закінчена програмна одиниця, яка структурно ідентифікується (або оформляється) стандартним чином по відношенню до компілятора і по відношенню до об'єднання її з іншими аналогічними одиницями в завантаженні. У загальному випадку будь-яка програма має зовнішні і внутрішні потоки даних, області зберігання даних. [60] Тому, як правило, модуль має ідентифікаційний документ, що містить інформацію про характеристики модуля: змінні, обсяг модуля, точки входу в модуль, параметри налаштування, мова програмування та інше. Звичайно модуль

не перевищує 1 тисячу команд комп'ютера або операторів мови програмування. Найбільш оптимальний за розміром модуль, як правило, зменшується на одному місці роздруківки на АЦДП. [59] В цьому випадку, модульна система, є найбільш технологічною.

Модульне програмування - це мистецтво розбиття завдання на декілька модулів та правила широкого використання стандартних модулів, автоматизації складання готових модулів з бібліотек, банків модулів та інше. [58] Дуже добре в модульному програмуванні використовується техніка макрогенерації.

Модульне програмування, має наступні переваги:

- простота складання і налагодження модульної програми, тому що функціональні компоненти програми можуть бути написані і налагоджені порізно;
- модульна програма легше супроводжується і модифікується, тому що функціональні компоненти можуть бути змінені, переписані або замінені, без зміни, в інших компонентах;
- полегшення процесу управління, розробкою модульної програми, тому що складніші модулі, які можна розробляти більш досвідченими програмістами, простіше здійснюється контроль і звітність програмістів.

Але, модульне програмування, має і недоліки:

1. Потрібно виконати великий обсяг робіт, тому що програміст повинен проектувати програми по низхідній схемі і аналізувати на кожному кроці декомпозиції програми, зручність її модифікації, а так само проводити документування програми на кожному кроці декомпозиції;
2. Зростають витрати часу центрального процесора на виконання програми і займається кілька більший обсяг пам'яті під програму (на ~ 5 -10%), тому що необхідно при вході і виході з підпрограми виконувати команди, які забезпечують їхню сумісність з програмою вищого рівня;
3. Можуть виникнути труднощі в системах реального часу, тому що при зчитуванні підпрограм в оперативну пам'ять, програма може бути переведена в режим очікування.

Існує кілька методів побудови модульних програм. Перший метод характеризується функціональною незалежністю модулів. Кожна підпрограма повинна мати певне призначення, яке більш-менш залежить від інших підпрограм програми. [57] Вибір певного розміру модуля, визначається розв'язуваною завданням, зручністю супроводу, мовою програмування. Якість розбиття програми на модулі, багато в чому залежить, від кваліфікації програміста.

Другий метод полягає в використанні, так званих таблиць рішень, які дозволяють формально записувати логічні алгоритми і реалізувати їх інваріантними програмами. [56] Цей метод, має важливі переваги:

- 1) сприяє кращому аналізу і розуміння завдання, тому що дисциплінує думки розробника алгоритму;
- 2) організовує зручний спосіб спілкування між розробниками алгоритмів і програмістами, що виключає багато неоднозначності, властиві словестного описами програмованої завдання;
- 3) зменшує, а в деяких випадках й повністю вимикає програмування алгоритмів;
- 4) забезпечує зручний контроль помилок, зменшує неповноту, суперечливість і надмірність алгоритмів;
- 5) створює можливість визначати модульність ПС і організувати базу знань розробнику алгоритмів (технологу).

Третій метод полягає в відділенні операцій по вводу / виводу від обчислювальних операцій. Це сприяє швидкій зміні та зручному завданням інтерфейсу між ПС і користувачем. При цьому обчислювальні операції не змінюються. [55] Крім цього, створюються передумови, для розробки інваріантного набору програм введення / виведення.”

Також для робробки системи необхідно використати техніку об'єктно-орієнтованого програмування.

Об'єктно-орієнтоване програмування (ООП) - методологія програмування, виживою основою якого є представленні програми як сукупність об'єктів , які виконують роль екземплярів певного класу.

Ідеологічно ООП - підхід до програмування як до моделювання інформаційних об'єктів, вирішальний на новому рівні основне завдання структурного програмування: структурування інформації з точки зору керованості, що істотно поліпшує керованість самим процесом моделювання, що в свою чергу особливо важливо при реалізації великих проектів.

Керованість для ієрархічних систем передбачає мінімізацію надмірності даних (аналогічну нормалізації) і їх цілісність, тому створене зручно керованим - буде і зручно розумітися. Таким чином через тактичну задачу керованості вирішується стратегічне завдання - транслювати розуміння завдання програмістом в найбільш зручну для подальшого використання форму.

Основні принципи структурування в разі ООП пов'язані з різними аспектами базового розуміння предметного завдання, яке потрібно для оптимального управління відповідною моделлю:

- 1) абстрагування для виділення в моделюємому предметі важливого для вирішення конкретного завдання по предмету, в кінцевому рахунку - контекстне розуміння предмета, формалізуються у вигляді класу;
- 2) інкапсуляція для швидкої і безпечної організації власне ієрархічної керованості: щоб було достатньо простої команди «що робити», без одночасного уточнення як саме робити, так як це вже інший рівень управління;
- 3) успадкування для швидкої і безпечної організації родинних понять: щоб було достатньо на кожному ієрархічному кроці враховувати тільки зміни, що не дублюючи все інше, враховане на попередніх кроках;
- 4) поліморфізм для визначення точки, в якій єдине управління краще розділити або навпаки - зібрати воедино.

Тобто фактично мова йде про прогресуючу організацію інформації згідно з первинними семантичним критеріям: «важливе / неважливе», «ключове / подробиці», «батьківське / дочірнє», «єдине / множинне». Прогресування, зокрема, на останньому етапі дає можливість переходу на наступний рівень деталізації, що замикає загальний процес.

Звичайна людська мова в цілому відображає ідеологію ООП, починаючи з інкапсуляції уявлення про предмет у вигляді його імені і закінчуючи поліморфізмом використання слова в переносному сенсі, що в підсумку розвиває вираження думки через ім'я предмета до повноцінного поняття-класу.

До основних понять ми можемо віднести:

1) абстрагування даних. Це означає виділення значущої інформації і виключення з розгляду незначною. В ООП розглядають лише абстракцію даних (нерідко називаючи її просто «абстракцією»), маючи на увазі набір значущих характеристик об'єкта, доступний решті програмою.

2) інкапсуляція - часто розглядається як поняття, властиве виключно об'єктно-орієнтованого програмування (ООП), але в дійсності широко зустрічається і в інших. Мови C++, Java або Ruby ототожнюють інкапсуляцію з приховуванням, тоді як інші (Smalltalk, Eiffel, OCaml) розрізняють ці поняття.

3) наслідування - концепція об'єктно-орієнтованого програмування, згідно з якою абстрактний тип даних може успадковувати дані і функціональність деякого існуючого типу, сприяючи повторному використанню компонентів програмного забезпечення.

4) поліморфізм - в об'єктно-орієнтованому програмуванні здатність об'єкта вибирати правильний метод в залежності від типу даних, отриманих в повідомленні. Поліморфізм в мовах програмування - поліморфізм (в мовах програмування) взаємозамінність об'єктів з однаковим інтерфейсом.

5) клас - універсальний, комплексний тип даних, що складається з тематично єдиного набору «полів» (змінних більш елементарних типів) і «методів» (функцій для роботи з цими полями), тобто він є моделлю інформаційної сутності з внутрішнім і зовнішнім інтерфейсами для оперування своїм вмістом (значеннями полів). Зокрема, в класах широко використовуються спеціальні блоки з одного або частіше двох спарених методів, що відповідають за елементарні операції з певним полем (інтерфейс присвоювання і зчитування значення), які імітують безпосередній доступ до поля. Ці блоки називаються «властивостями» і майже збігаються по конкретному імені зі своїм полем (наприклад, ім'я поля може починатися з малої, а

ім'я властивості - з великої літери). Іншим проявом інтерфейсної природи класу є те, що при копіюванні відповідної змінної через присвоювання, копіюється тільки інтерфейс, але не самі дані, тобто клас - контрольний тип даних. Змінна-об'єкт, що відноситься до заданого класом типу, називається екземпляром поточного. В деяких системах клас також може бути деяким об'єктом при виконанні програми в наслідок ідентифікації типу даних. В нормальних умовах класи розробляють так, щоб забезпечити відповідність природі об'єкту і розв'язуваної задачі цілісність даних об'єкта, а також зручний і простий інтерфейс. У свою чергу, цілісність предметної області об'єктів і їх інтерфейсів, а також зручність їх проектування, забезпечується успадкуванням.

5.5. Робота користувача з програмою

У цьому розділі описані вимоги до обладнання та програмного забезпечення для роботи з програмою, а також базові сценарії роботи користувача з програмою. Нижче наведені вимоги виконуються на більшості сучасних персональних комп'ютерів, що не обмежує коло приватних осіб і організацій, що можуть використовувати програмний продукт.

Інсталяція та системні вимоги. Для використання програмного забезпечення користувачу потрібний браузер з підтримкою WebGL. До таких браузерів належать: Google Chrome 9+, Firefox 4+, Opera 15+, Safari 5.1+, Internet Explorer 11 та Microsoft Edge. Робота програмного забезпечення не залежить від пристрою, це може бути комп'ютер з Microsoft Windows, Linux чи MacOS або будь-який мобільний пристрій, що підтримує актуальні версії браузерів.

При запуску програмного застосунку користувач спочатку бачить вікно авторизації, яке зображене на рисунку 5.2.

Рисунок 5.3 — Головне вікно

Після успішної авторизації користувач потрапляє в головне вікно програмного застосунку, що зображено на рисунку 5.3.



Рисунок 5.3 — Головне вікно

Чорна область є робочою зоною, де відбувається побудова, та деформація геометричних об'єктів. Інші маніпуляції з системою відбуваються за допомогою меню, що зображено на рисунку 5.4. Елементи управління представлені в меню будуть розглянуті далі.

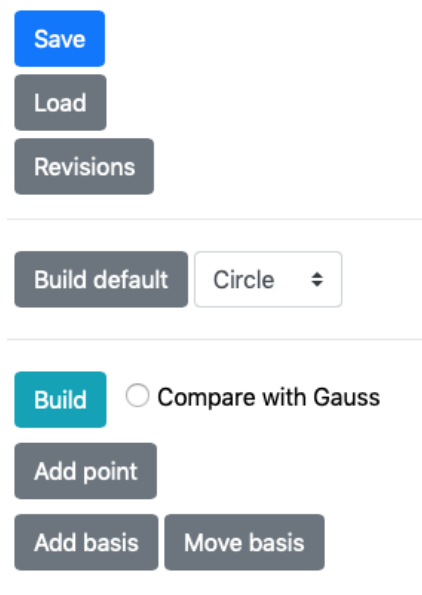


Рисунок 5.4 — Меню системи моделювання поцесів деформації геометричних об'єктів

Щоб додати новий геометричний об'єкт потрібно спочатку побудувати його за допомогою точок, що зображено на риснку 5.5.

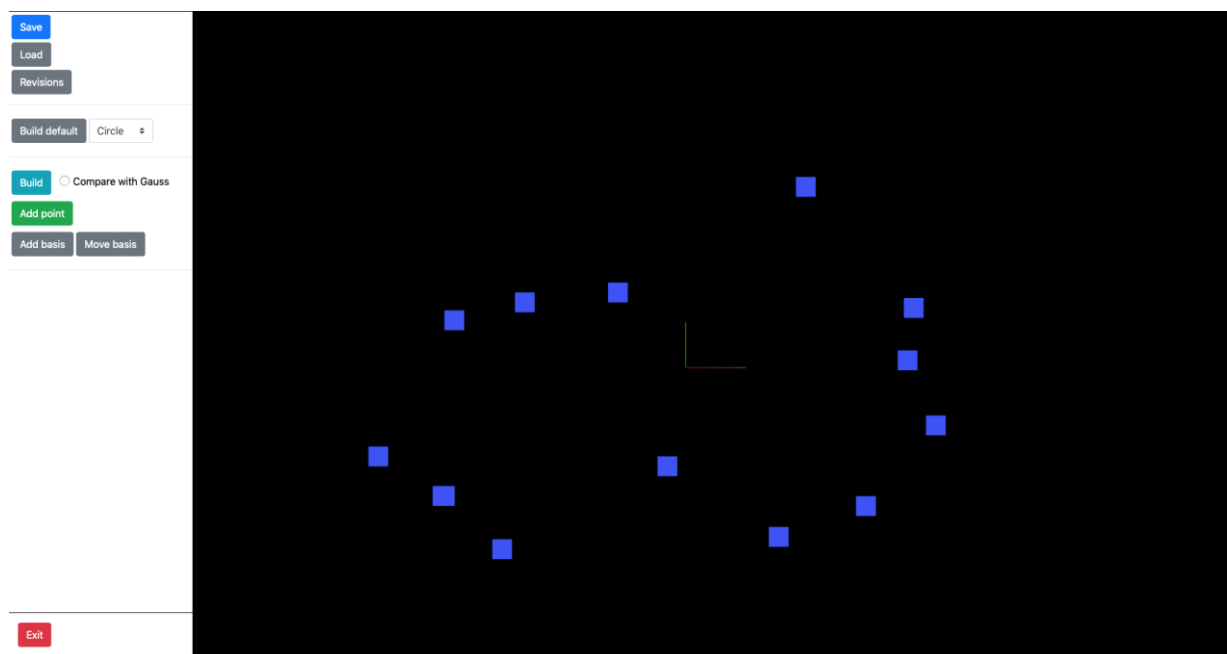


Рисунок 5.5 — Встановлення точок

Після натискання на кнопку “Add point”, вона стає активною і дозволяє користувачеві встановити точку в робочій зоні. Після цього кнопка стає неактивною.

Для того, щоб додати нову точку потрібно повторити процедуру. Після встановлення точок геометричного об'єкта, що буде деформуватися, потрібно встановити точки базисів, натиснувши на кнопку в меню “Add basis”. Після цього кнопка стає активною і дозволяє встановлювати точки базисів при натисканні в подрібному місці в робочій зоні. Базиси відображаються зеленими точками, більшого розміру ніж опорні точки об'єкту деформації. Процес встановлення точок базисів зображений на рисунку 5.6. Щоб побудувати геометричний об'єкт потрібно натиснути кнопку “Build” в меню. Готовий геометричний об'єкт, що буде деформуватися, зображений на рисунку 5.7. Потрібно звернути увагу, що після побудови геометричного об'єкта під меню з'явилась інформація про об'єкт. Це панель інформації про поточний стан системи. Після деформування об'єкта в ній з'явиться додаткова інформація, в залежності від налаштувань.

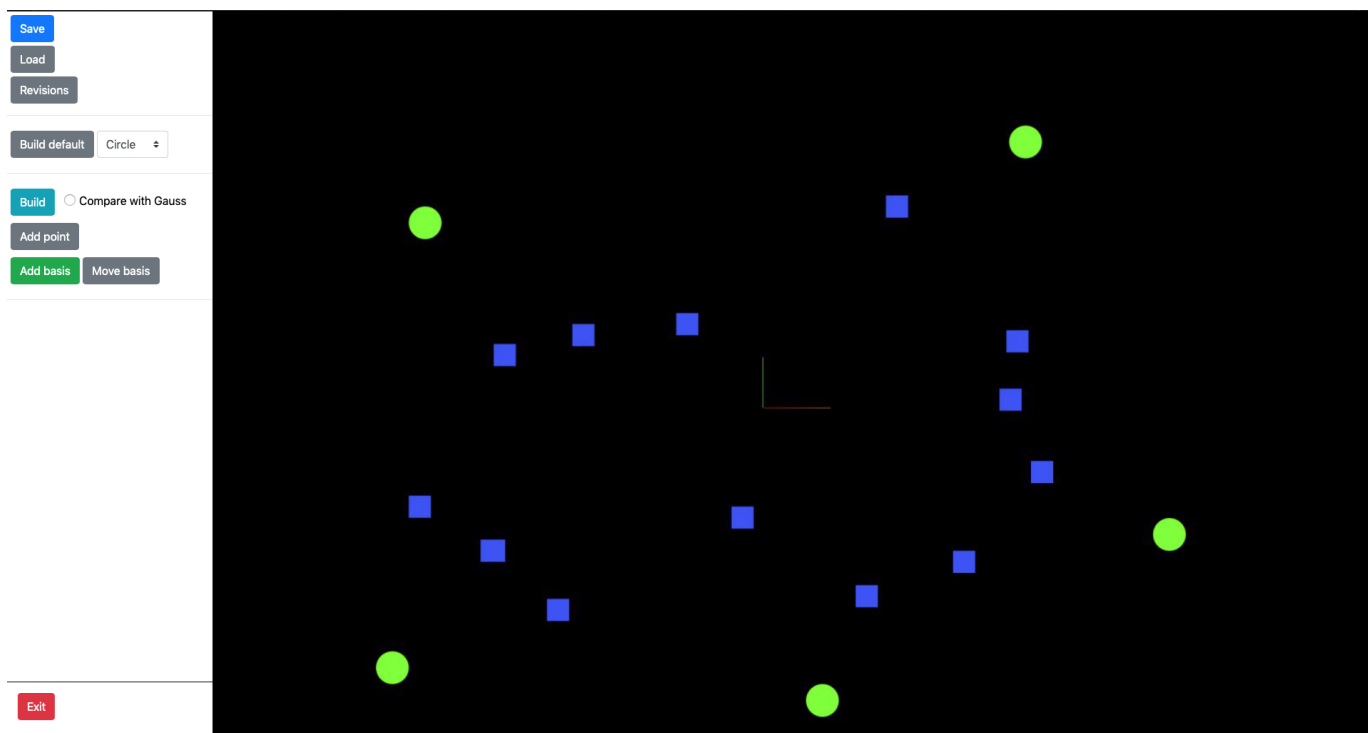


Рисунок 5.6 — Встановлення точок базисів

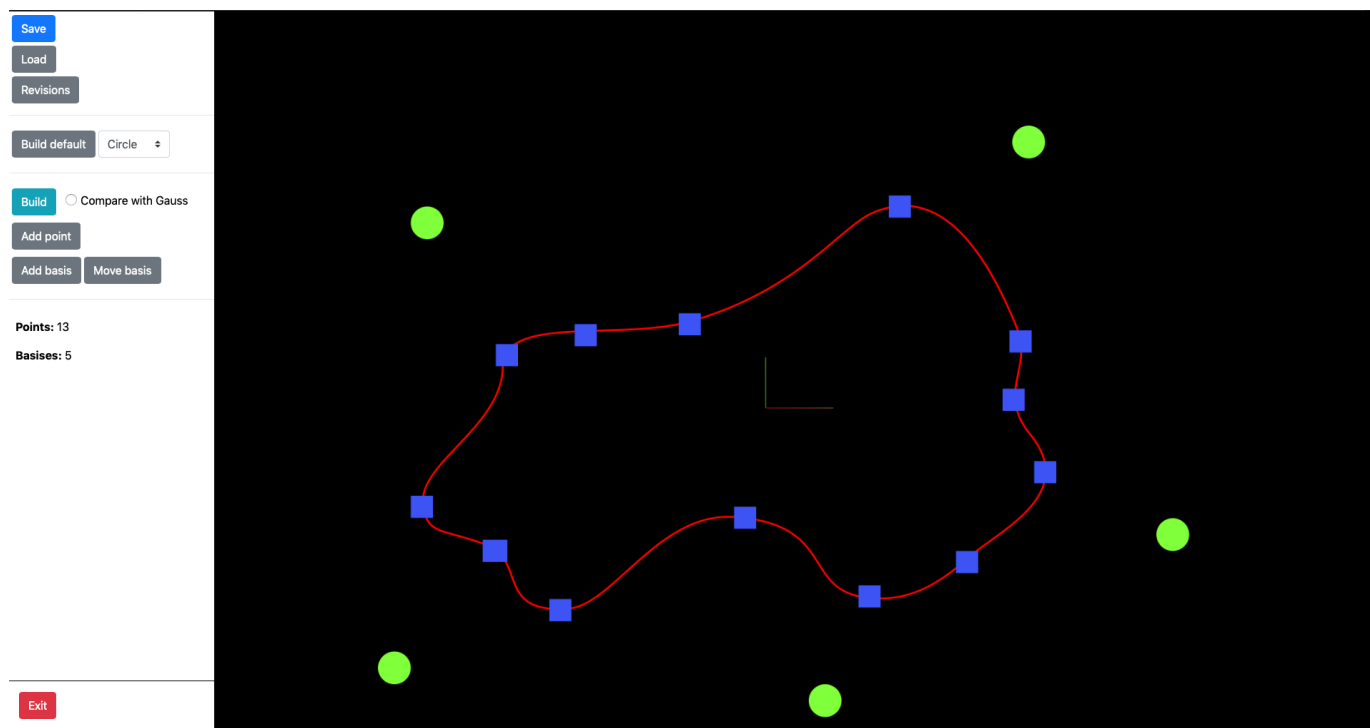


Рисунок 5.7 — Геометричний об’єкт

Також користувач має можливість пропустити процес встановлення опорних точок геометричного об’єкта та точок базисів і побудувати відразу об’єкт за замовчуванням кнопкою “Build default”. Користувачу доступні коло, трикутник та квадрат в якості стандартних геометричних об’єктів. Змінити об’єкт за замовчуванням можна за допомогою селекту, праворуч від кнопки “Build default”. На рисунку 5.8 зображено результат генерації за замовчуванням, а саме коло. Майбутні операції будемо проводити з отриманим об’єктом. Також можемо звернути увагу, що в панелі інформації появилось додаткове поле з повідомленням “Default object: circle”, що побудовано об’єкт за замовчуванням, а саме коло.

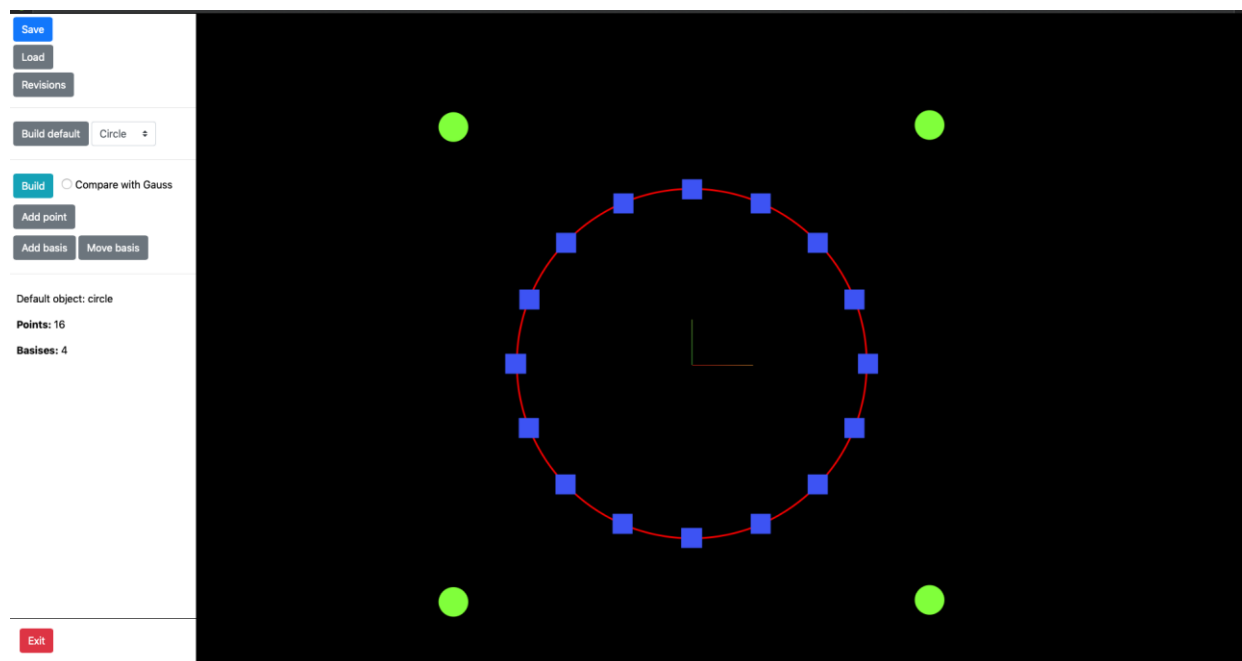


Рисунок 5.8 — Геометричний об’єкт за заамовчуванням, а саме коло

Для деформації геометричного об’єкта потрібно перейти в режим переносу точок базисів. Для цього потрібно натиснути кнопку “Move basis”, після чого кнопка стане активною і користувач має можливість пересувати точки базисів. Для того, щоб пересунути точку, потрібно натиснути на неї, після чого базис виділиться жовтим кольором, і натиснути на нове положення точки базису на робочій зоні. Для того, щоб провести також деформацію за допомогою функції Гауса та порівняти результати з областю створену в процесі політочкових перетворень, потрібно увімкнути кнопку в меню “Compare with Gauss”, що знаходиться праворуч від кнопки “Build”. Після переносу точки базису деформація геометричного об’єкту відбудеться автоматично. Це зображено на рисунку 5.9.

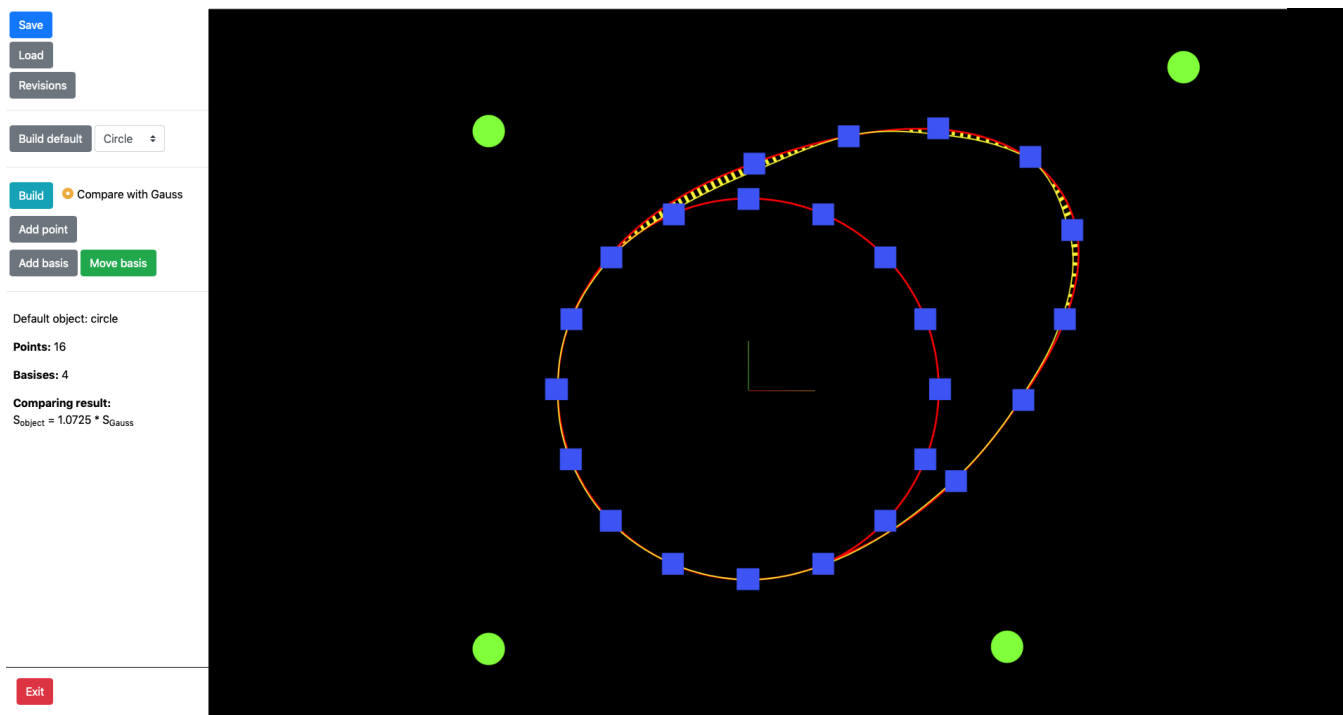


Рисунок 5.9 — Деформований геометричний об'єкт

Після деформації геометричного об'єкту в панелі інформації з'являється результати порівняння проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса. А саме відображається співвідношення площі об'єкта створеного проведенням політочкових перетворень області до площі об'єкта заданого функцією Гауса.

Після закінчення роботи з об'єктом користувач може зберегти його. Для цього потрібно натиснути кнопку “Save” і ввести ім'я об'єкта. Процес збереження об'єкта зображено на рисунку 5.10. Після збереження користувач може завантажити об'єкт для провадження роботи з ним або з раніше збереженим об'єктом. Для цього потрібно натиснути кнопку в меню “Load”. Після цього з'явиться спливаюче вікно зі списком збережених геометричних об'єктів, що зображено на рисунку 5.11. Щоб продовжити роботу об'єктом користувачу потрібно натиснути кнопку “Open” навпроти потрібного. Також користувач має можливість видалити об'єкт, якщо він більше не збирається з ним працювати. Для цього потрібно натиснути кнопку “Delete”, проте слід звернути увагу на те, що система немає можливості відновити видалені об'єкти.

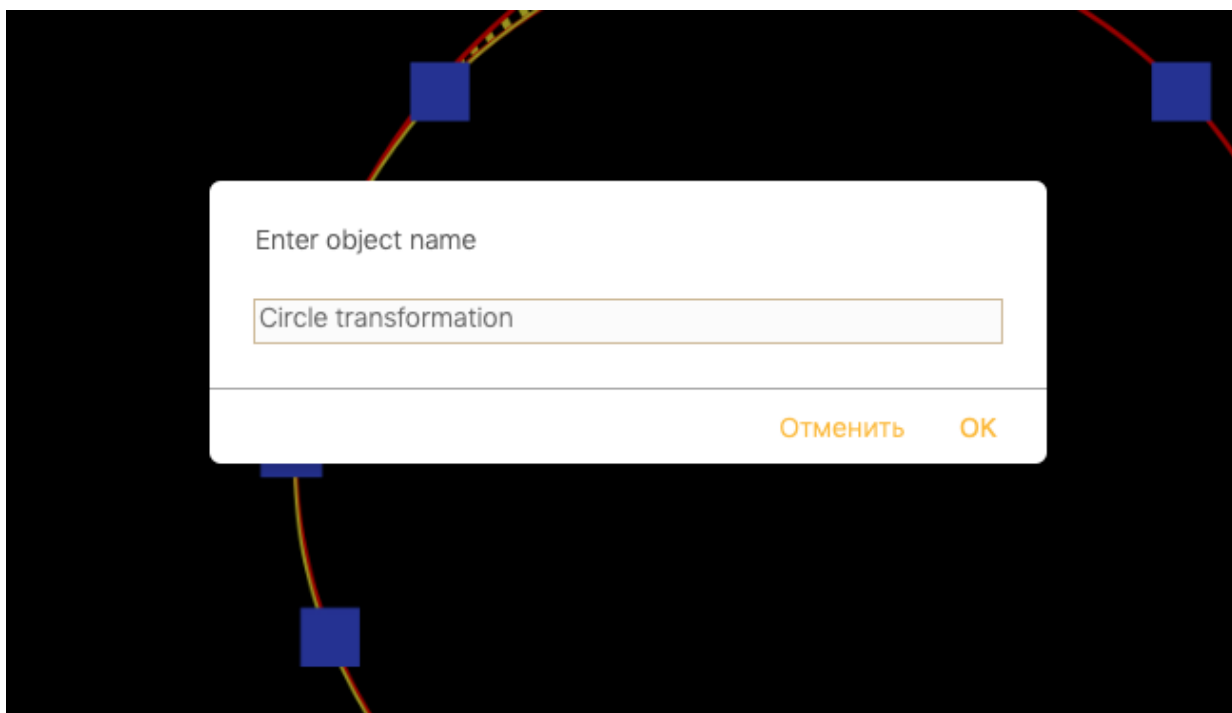


Рисунок 5.10 — Вікно збереження об'єкта

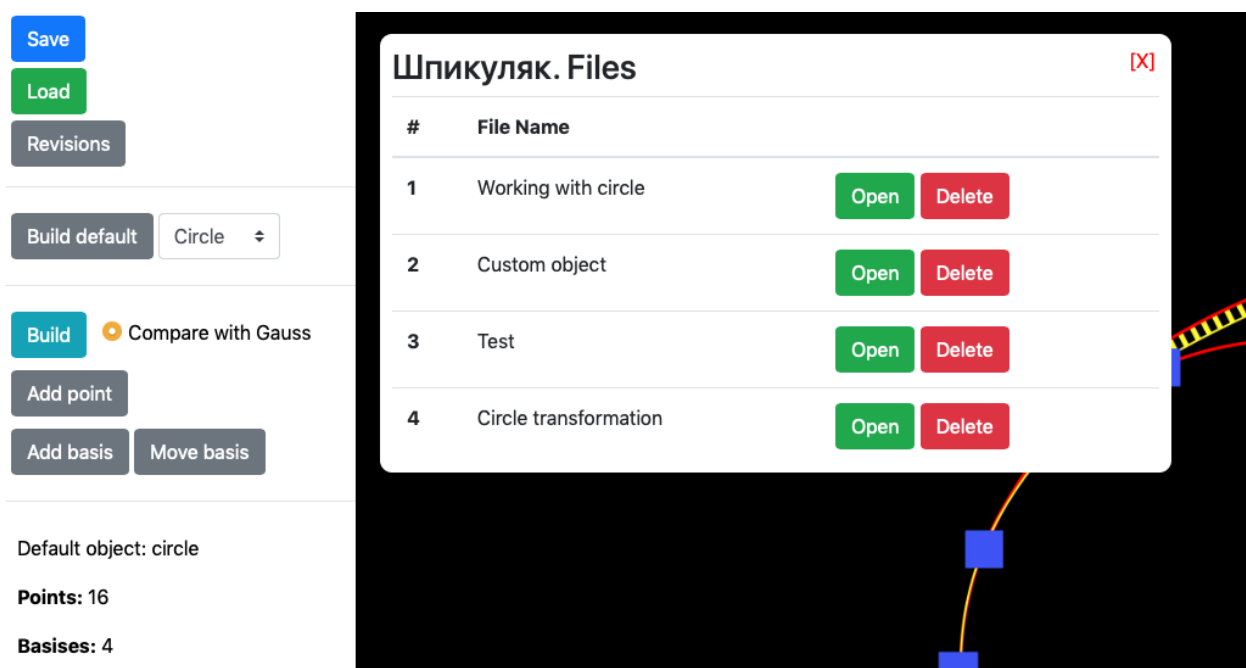


Рисунок 5.11 — Вікно збереження об'єкта

Для завершення роботи з аккаунтом користувачу потрібно вийти з нього, натиснувши кнопку “Exit”. Користувач може повернутись до роботи в будь-який час та з будь-якого комп'ютера або мобільного пристрою з доступом в інтернет і браузером, що підтримує технологію WebGL.

Отже, можна виділити основні положення розділу:

- Перед початком розробки системи було побудовано її функціональну модель, що складається з інтерфейсного модуля, розрахункового модуля та модуля візуалізації.
- Розрахунковий модуль відповідає за проведення обрахунків системою, інтерфейсний модуль забезпечує механізми введення інформації для обробки програмою. Модуль візуалізації відповідальний за відображення інформації користувачу.
- Для використання програмного забезпечення користувачу потрібний браузер з підтримкою WebGL. До таких браузерів належать: Google Chrome 9+, Firefox 4+, Opera 15+, Safari 5.1+, Internet Explorer 11 та Microsoft Edge. Робота програмного забезпечення не залежить від пристрою, це може бути комп'ютер з Microsoft Windows, Linux чи MacOS або будь-який мобільний пристрій, що підтримує актуальні версії браузерів.
- Методика роботи користувача з системою моделювання процесів деформації геометричних об'єктів зводиться до роботи з системою екранних форм та меню.
- До найбільш суттєвого наукового висновку цього розділу можна віднести те, що набуло подальшого розвитку використання методів геометричного моделювання для відображення зміни форми об'єкта в реальному часі.

6. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Оскільки дана робота несе конгитивний характер, то в основну ідею стартап-проекту покладено створення програмної системи, що надає можливість швидкого реагування на виникаючі лісові пожежі. Також система може будувати прогноз розповсюдженні пожежі, може виводити найближчі населені пункти з номерами телефонів адміністрації, може надавати можливість змінювати координати з наступною перебудовою прогнозу у відповідності з новими даними, може підраховувати збитки в кожний момент часу. Програмне забезпечення може бути впроваджено в технологічний комплекс підрозділів міністерства надзвичайних ситуацій або підприємствам, відповідальним за стан об'єктів господарювання.

6.1. Опис ідеї стартап-проекту

Проаналізуємо значення ідеї, її можливі напрямки прикладання, чим дана ідея відрізняється від існуючих аналогів, а також основні плюси, які може отримати користувач програмної системи. Результати аналізу представлені у таблиці 6.1.

Таблиця 6.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення системи моделювання та прогнозування розповсюдження лісової пожежі на основі полікоординатних перетворень	1. Моделювання пожеж у лісі	Своєчасна достовірна модель пожеж в лісі
	2. Прогнозування розповсюдження пожеж у лісі	Оцінка потенційних збитків від пожежі в лісі, своєчасне попередження населених пунктів

Зараз не існує програмного продукту, що фіксує та зберігає інформацію про лісову пожежу, будує прогноз її поширення, надає можливість редагування цього

прогнозу з майбутнім його корегуванням, визначає найбільш уразливі населені пункти, куди треба організувати маршрути пересування техніки для гасіння пожежі та для відселення населення визначених територій, розраховує втрати на ураженій території. Деякий функціонал було реалізовано сторонніми утилітами, але використання його нефункціональне та неоптимальне.

Аналіз потенційних техніко-економічних переваг ідеї порівняно з пропозиціями конкурентів проводити не доцільно.

6.2. Технологічний аудит ідеї проекту

Для проведення технічного аудиту значення проекту, потрібно провести аудит технології, за допомогою якої можна реалізувати ідею проекту. І для початку потрібно визначити, що проект можливо технологічно здійснити. Результат представлений у таблиці 6.2.

Таблиця 6.2 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Створення системи моделювання та прогнозування розповсюдження пожежі в лісі за допомогою полікоординатних перетворень	Середовище розробки файлів-застосунків Sublime Text	✓	Доступно (безкоштовне)
	Технологія WebGL	✓	Доступно (безкоштовне)

Обрана технологія реалізації ідеї проекту: Sublime Text – середовище розробки файлів-застосунків та використання технології WebGL для побудови моделі розповсюдження лісових пожеж на реальній карті світу.

Обрана технологія доступна, не потребує доробки, а також безкоштовна та надає усі необхідні можливості для реалізації поставленої задачі.

6.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, що можливо використати під час ринкового впровадження проекту, та ринкових небезпек, які можуть перешкодити реалізації проекту, дозволяє спланувати шляхи розвитку проекту з врахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Для цього спочатку проведемо аналіз попиту (таблиця 6.3).

Таблиця 6.3 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку	Характеристика
Загальна потреба в продукції	Необхідна, але багатьма не признається (через фінансові вигоди)
Можливі річні обсяги випуску в натуральних показниках	До 1000 копій
Ціна одиниці продукції	200\$
Річні обсяги випуску в вартісних показниках	20000 – 200000\$
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу	Бажання розробників працювати лише над власним ПЗ, задля підтримки монополії у сфері
Специфічні вимоги до стандартизації та сертифікації	Для ПЗ відсутні.
Середня норма рентабельності в галузі (або по ринку)	76%

Враховуючи значну необхідність у розробці подібних програмних застосунків, задля покращення процесів розробки програмних продуктів та полегшення праці розробникам, доцільно розробляти дану систему.

Визначення ринкових можливостей, що можливо використати під час ринкового впровадження проекту, та ринкових небезпек, що можуть перешкодити реалізації даного проекту, дозволяє спланувати шляхи розвитку проекту з

урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Після того, як визначились потенційні групи клієнтів, проводиться аналіз ринкового середовища: складаються таблиці факторів, які слугують ринковому впровадженню проекту, та факторів, що йому перешкоджають.

Результати представлені у таблицях 6.4 та 6.5 відповідно.

Таблиця 6.4 – Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Поява конкурентів	Можлива поява конкурентів, що спроможуться створити більш якісний продукт. Можлива поява більш дешевих продуктів	Зменшення ціни на продукт з підвищенням якості при цьому, розробка удосконалень, розширення асортименту (додавання нових можливостей, нового функціоналу або додання можливостей розрахунку нових параметрів)
Зміни тенденцій ринку	Можлива ситуація, в якій появиться більш досконала програмна система від конкурентів, що значно довше на ринку.	Майже неможлива ситуація на найближчі роки. Але можливості вирішення доволі прості - розробка нових сучасних та необхідних удосконалень, тобто додання або рефакторинг старого функціоналу для можливості розрахунку нових параметрів
Зниження репутації компанії	Можлива ситуація, якщо конкуренти спроможуться на більший попит	Зміна партнерів, підписання нових контрактів, проведення рекламних або промо-акцій
Економічний спад	Відсутність попиту на товар компанії через економічну складову	Зміна цільової аудиторії; збільшення об'ємів продажів, зменшення ціни;

Таблиця 6.5 – Фактори можливостей

Фактор	Зміст загрози	Можлива реакція компанії
--------	---------------	--------------------------

Невелика кількість конкурентів	На ринку на даний момент дуже не значна кількість конкурентів, у яких програмні продукти в переважній більшості гірші або вузько спеціалізовані	Розповсюджувати продукт, розвивати його можливості
Відповідні тенденцій ринку	ІТ-ринок на сьогоднішній день потребує, а відповідно і надає всі можливості для впровадження систем, що будуть надавати користувачам можливість моделювання лісових пожеж.	Розповсюджувати продукт, розвивати його можливості
Можливість побудови власної репутації	Новий «гравець» на ринку має всі можливості для побудови власної репутації з «чистого листка»	Пошук замовників, можливих покупців створеного продукту, розширення бази замовників. Зарекомендувати себе, як надійну компанію. Можливо на вигідних умовах співпраці

Надалі проводиться аналіз пропозиції – визначаються загальні риси конкуренції на ринку (таблиця 6.6): визначається тип можливої майбутньої конкуренції та її інтенсивність, рівень конкурентоспроможності за рівнем конкурентної боротьби, галузевою ознакою і видами товарів.

Таблиця 6.6 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	У чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)
Тип конкуренції	Чиста залежить від кількості конкурентів та якості надання ними послуг у порівнянні з послугами компанії	Покращення продукту за допомогою зниження ціни і підвищення якості
За рівнем конкурентної боротьби	Локальна Конкуренція на вітчизняному ринку	На вітчизняному ринку конкурентів не виявлено, а тому компанія має можливість встановлення власної ціни,

		та набирати клієнтську базу. Перспектива – вихід на міжнародний ринок.
За галузевою ознакою	Внутрішньогалузева Продукт націлений лише на конкретну сферу діяльності	Немає можливостей та сенсу поширювати функціонал за межі ІТ-сфери, проте існує багато варіантів розвиватись всередині неї
Конкуренція за видами товарів	Марки-конкуренти Створений товар може мати конкурентів, які пропонують аналогічний товар	Розширення функціональних, безкоштовне встановлення в інститутах (зادля популяризації підходу)
За характером конкурентних переваг	Цінова Важливо за скільки продається товар, та скільки з нього прибутку	Можливе підвищення ціни на нові розробки, зниження на старі версії для заохочення покупців у порівнянні з цінами конкурентів
За інтенсивністю	Марочна Можуть з'являться конкуренти	На ринку цільової аудиторії поки що конкурентів не виявлено. Але при виході на міжнародний ринок потрібно рекламувати кращий функціонал створеного продукту, встановлювати конкурентоспроможні ціни, та доводити свою надійність

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 6.7) - за моделлю п'яти сил М. Портера, яка вирізняє п'ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції:

- конкурент, що вже є у галузі;
- потенційні конкуренти;
- наявність товарів-замінників;
- постачальники, що конкурують за ринкову владу;
- споживачі, які конкурують за ринкову владу.

Аналіз оформлюється за таблицею 6.7, обґрунтування факторів за таблицею 6.8.

Таблиця 6.7 – Аналіз конкуренції в галузі за М.Портером

Складові галузі	Прямі конкуренти в галузі	Потенційні конкуренти	Клієнти	Товари-замінники
	Розробники аналогічних систем	Кращі продукти, менші ціни	Мають найбільше значення. Більш важлива їх кількість, аніж постійна співпраця	Відсутні. Є лише конкуренти аналогічних розробок
Висновки	Інтенсивність конкурентної боротьби з боку прямих конкурентів незначна	Наявні усі можливості входу на ринок. Потенційні конкуренти не виявлені. Строки виходу на ринок – один день	Необхідність клієнтської бази, тому важливо знаходити можливості приваблення споживачів до власного продукту	Немає обмежень

Таблиця 6.8 – Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Невелика кількість конкурентів на ринку	На вітчизняному ринку, на який для старту націлена розроблена система, конкурентів немає
Доступність створеного продукту (програмно)	Немає жорстких системних вимог, програма буде працювати навіть на застарілих ПК

Таблиця 6.8 (продовження)

Фактор конкурентоспроможності	Обґрунтування
Легкість і простота використання	Зручний зрозумілий інтерфейс, створені довідка та інструкція для користувача
Відсутня потреба у постійному супроводі	Не потребує супроводу спеціалістів і постійних доробок з боку розробника
Підключення до мережі Інтернет	Потребує підключення до мережі Інтернет після придбання продукту
Додаткові компоненти	Немає необхідності встановлення додаткових компонентів, на відміну від деяких аналогів, які не працюють без АПК

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту, проведений у таблиці 6.9.

Таблиця 6.9 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
		-3	-2	-1	0	+1	+2	+3
Мала кількість / відсутність конкурентів	10				✓			
Системні вимоги	18				✓			
Простота використання	18	✓						
Не потрібен супровід	10					✓		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.10) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін.

Таблиця 6.10 – SWOT-аналіз проекту

Сильні сторони (S):	Слабкі сторони (W):
<ul style="list-style-type: none"> – невелика кількість працівників; – молодий і перспективний колектив; 	<ul style="list-style-type: none"> – брак власного устаткування; – брак робочої сили;

<ul style="list-style-type: none"> – гнучка політика керівництва; – інноваційні технології 	<ul style="list-style-type: none"> – недостатньо оборотних коштів; – відсутність репутації компанії;
<p>Можливості (О):</p> <ul style="list-style-type: none"> – розширення виробничої лінії; – додаткові послуги; – вихід на нові ринки; – розширення клієнтської бази; – співробітництво з іншими компаніями 	<p>Загрози (Т):</p> <ul style="list-style-type: none"> – поява нових конкурентів; – зміни тенденцій попиту; – зниження репутації компанії; – економічний спад

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії. Основна цільова аудиторія – це дослідницькі та будь-які інші установи, що займаються моделюванням та прогнозуванням розповсюдження лісової пожежі.

Стан та динаміка ринкового середовища на сьогоднішній день і ще багато років є і будуть залишатись сприятливими для впровадження розробленої системи.

ВИСНОВКИ

- Аналіз систем моделювання та деформації геометричних об'єктів показав, що існує необхідність в реалізації системи що дасть змогу досліджувати метод політочкових перетворень, що дає змогу побудувати систему з можливістю зорового відстеження результатів проведення політочкових перетворень та порівняти результати проведення політочкових перетворень певної області з областю заданою за допомогою функції Гауса. Таким чином, актуальною є розробка та вдосконалення комп'ютерних технологій геометричного моделювання.

- На основі аналізу предметної області та визначення мети роботи, що полягає створенні системи моделювання процесів деформації геометричних об'єктів, виявленні нових підходів до деформаційного моделювання за допомогою удосконалення методів геометричного моделювання, були сформульовані завдання досліджень:

- проаналізувати існуючі методи геометричного моделювання об'єктів;
- дослідити особливості та визначити властивості полікоординатних перетворень;
- розробити програмний продукт для візуалізації процесу динаміки зміни крайки вигорання лісу при пожежі;

- з точки зору особливостей поставленої проблеми для визначення можливостей вирішення задачі проведено аналіз існуючих методів і систем;

- проаналізовано недоліки цих методів та обґрунтована необхідність їх модифікації;

- виділені напрямки досліджень для вирішення задачі моделювання процесів деформації геометричних об'єктів на основі полікоординатних перетворень.

- Наведено математичний апарат системи моделювання процесів деформації геометричних об'єктів на основі полікоординатних перетворень, а саме:

- апарат політочкових перетворень;
- різновиди політочкових перетворень;

Для розробки програмної системи було обрано такий інструментарій:

- мова програмування JavaScript;
- технологія WebGL;
- бібліотека Three.js;
- мова програмування PHP;
- фреймворк Laravel;
- база даних MS SQL Server;

- Було побудовано функціональну модель, що складається з інтерфейсного модуля, розрахункового модуля та модуля візуалізації.

- Набуло подальшого розвитку використання методів геометричного моделювання для відображення зміни форми об'єкта в реальному часі.

- Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії. Основна цільова аудиторія – це дослідницькі та будь-які інші установи, що займаються моделюванням та прогнозуванням розповсюдження лісової пожежі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аушева Н.М. Моделювання гладких поверхонь при екологічних розрахунках [Текст] / Н.М. Аушева // Прикладна геометрія та інженерна графіка. – К.: КДТУБА, 1998. – Вип. 63. – С. 217-219.
2. Ашик В.В. Проектирование судов / В.В. Ашик. – Л.: Судостроение, 1985. – 320 с.
3. Бабаков В.В. Проектирование поверхностей кривыми второго порядка в самолетостроении [Текст] / В.В. Бабаков. – М.: Машиностроение, 1969. – 124 с.
4. Бадаєв С.Ю. Проектування криволінійних обводів за заданим законом кривини [Текст] / С.Ю. Бадаєв // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2002. – Т. 15. – С. 93-96.
5. Бадаєв С.Ю. Криволінійний обвод за заданим законом кривини на основі колового сплайну [Текст] / С.Ю. Бадаєв // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2002. – Вип. 71. – С. 172-177.
6. Бадаєв Ю.И. Полилинейные и нелинейные геометрические преобразования и алгебраические кривые [Текст] / Ю.И. Бадаєв // Прикладная геометрия и инженерная графика. – К.: Будівельник, 1987. – Вып. 44. – С. 32-35.
7. Бадаєв Ю.И. Поликоординатный метод в прикладной геометрии и компьютерной графике [Текст]: [науч. изд.] / Ю.И. Бадаєв – К.: Просвіта, 2006. – 172 с.
8. Бадаєв Ю.І. Криві на основі політканинних В-сплайнів [Текст] / Ю.І. Бадаєв // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2004. – Т. 28. – С. 16-20.
9. Бадаєв Ю.І. Апроксимація кривих на площині методом полікоординатних перетворень [Текст] / Ю.І. Бадаєв, Л.П. Лагодіна // Геометричне та комп'ютерне моделювання: зб. наук. праць. – Х.: ХДУХТ, 2007. – Вип. 17. – С. 103-107.

10. Бадаєв Ю.І. Проектування каналових поверхонь на основі політканинних перетворень [Текст] / Ю.І. Бадаєв, І.В. Овчарук // Водний транспорт. – К.: КДАВТ, 2001. – Вип. 2. – С. 168-173.

11. Бадаєв Ю.І. Політканинні перетворення в точковому визначенні [Текст] / Ю.І. Бадаєв, Ю.В. Сидоренко // Труды Таврической государственной агротехнической академии. – Вип. 4 : Прикладная геометрия и инженерная графика. – Мелітополь: ТГАТА, 1999. – Т. 8. – С. 21-23.

12. Бадаєв Ю.І. Конструювання геометричних об'єктів засобами політочкових перетворень [Текст] / Ю.І. Бадаєв, Ю.В. Сидоренко // Прикладна геометрія та інженерна графіка. – К.: КДТУБА, 2000. – Вип. 66. – С. 44-47.

13. Бадаєв Ю.І. Деформаційне конструювання об'єктів водного транспорту за допомогою політочкових перетворень [Текст] / Ю.І. Бадаєв, Ю.В. Сидоренко // Водний транспорт. – К: КДАВТ, 2000. – С. 140-143.

14. Бадаєв Ю.І. Врахування вагових коефіцієнтів при моделюванні процесу розтікання нафтової плями методом полікоординатного перетворення [Текст] / Ю.І. Бадаєв, Л.С. Чорна // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2004. – Т. 24. – С.42-45.

15. Бадаєв Ю. І. Властивості полікоординатних відображень за однією координатою / Ю. І. Бадаєв, Л. П. Лагодіна // Водний транспорт. - 2013. - Вип. 1. - С. 162-168. - Режим доступу: http://nbuv.gov.ua/UJRN/Vodt_2013_1_33.

16. Бадаєв Ю.І. Проектування параметричних кривих і поверхонь за допомогою політканинних перетворень [Текст] / Ю.І. Бадаєв, Л.С. Чорна // Проблеми сучасного підручника: зб. наук. праць. – К.: Педагогічна думка, 2004. – Вип. 5. – Ч. II. – С. 6–11. [64]

17. Бадаєв Ю.І. Полікоординатні векторно-параметричні криві на площині [Текст] / Ю.І. Бадаєв, Л.С. Чорна // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2006. – Т. 19. – С. 25-28.

18. Бадаєва Н.І. Евольвенти вищих порядків [Текст] / Н.І. Бадаєва // Труды Таврической государственной агротехнической академии. – Вып. 4 : Прикладная геометрия и инженерная графика. – Мелитополь: ТГАТА, 1999. – Т. 9. – С. 63-64.
19. Бадаєва Н.І. Евольвенти третього порядку [Текст] / Н.І. Бадаєва // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2002. – Вип. 71. – С. 153-155.
20. Бляшке В. Введение в геометрию тканей [Текст] / В. Бляшке; пер. с нем. М.А. Акивиса; под ред. И.М. Яглома. – М.: Физматгиз, 1959. – 144 с.
21. Бровченко И.А., Мадерич В.С. Численный лагранжевый метод моделирования распространения поверхностных пятен нефти // Прикладна гідромеханіка, 2002, т. 4(76), № 4, с. 1-9.
22. Ванін В.В. Побудова лінійного обводу, складеного із просторових дуг кривих четвертого порядку, по його графічно заданим проекціям [Текст] / В.В. Ванін, О.В. Павлов // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2002. – Т. 15. – С. 39-42.
23. Верещага В.М. Загальний метод згущення точкового ряду ДПК сум [Текст] / В.М. Верещага // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2002. – Вип. 71. – С. 63-67.
24. Вертинская Н.Д. Моделирование и конструирование поверхностей, несущих каркасы кривых высших порядков [Текст] / Н.Д. Вертинская // Современные проблемы геометрического моделирования: матер. Второй укр.-рос. науч.-практ. конф., 24-27 апреля 2007 г. : спецвып. – Х.: ХГУПТ, 2007. – С. 243-249.
25. Денискин Ю.И. Задание обводов летательных аппаратов полиномом Безье [Текст] / Ю.И. Денискин, В.И. Якунин // Геометрическое моделирование в авиационном и аэродромном проектировании: сб. науч. трудов. – К.: КИИГА, 1993. – С. 28-31.
26. Дорошенко Ю.О. Політканинні перетворення у керованому синтезі робочих поверхонь розпушувальних елементів без відвальної обробки ґрунту [Текст] / Ю.О. Дорошенко // Труды Таврической государственной агротехнической

академии. – Вып. 4 : Прикладная геометрия и инженерная графика. – Мелитополь: ТГАТА, 1997. – Т. 1. – С. 57-60.

27. Дорошенко Ю.О. Варіанти конструювання кінематичних поверхонь із застосуванням політканинних перетворень [Текст] / Ю.О. Дорошенко // Труды Таврической государственной агротехнической академии. – Вып. 4 : Прикладная геометрия и инженерная графика. – Мелитополь, ТГАТА, 1998. – Т. 3. – С. 55-58.

28. Дорошенко Ю.О. Політканинні перетворення у деформативному конструюванні геометричних об'єктів [Текст]: монографія / Ю.О. Дорошенко. – К.: Педагогічна думка, 2001. – 390 с.

29. Дорошенко Ю.О. Геометричне моделювання та корекція форми робочої поверхні ґрунтообробних знарядь відповідно до їх функціонального призначення гладкості [Текст] / Ю.О. Дорошенко // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2001. – Вип. 69. – С. 70-74.

30. Дорошенко Ю.О. Бінаправлена ортогональна деформація площини із застосуванням політканинних перетворень [Текст] / Ю.О. Дорошенко // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2003. – Вип. 72. – С. 87-93.

31. Ковалев С.Н. Дискретная интерполяция псевдоспиральными кривыми [Текст] / С.Н. Ковалев, О.Н. Ковтун // Современные проблемы геометрического моделирования: междунар. науч.-практ. конф.: тезисы докл. – Донецк: ДонГТУ, 2000. – С. 73-74.

32. Лагодіна Л.П. Моделювання кінематичної поверхні методом полікоординатних перетворень з урахуванням вагових коефіцієнтів [Текст] / Л.П. Лагодіна // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТУ, 2008. – Т. 39. – С. 123-127.

33. Маценко С.В., Волков Г.Г., Волкова Т.А. Ликвидация разливов нефти и нефтепродуктов на море и внутренних акваториях. Расчет достаточности сил и средств/ С.В. Маценко, Г.Г. Волков, Т.А. Волкова // Методические рекомендации. – Новороссийск, 2009. - с. 24-31

34. Михайленко В.Є. Дискретне геометричне моделювання: виникнення, результати, підсумки [Текст] / В.Є. Михайленко, С.М. Ковальов // Прикладна геометрія та інженерна графіка. – К, 2004. – Вип.74. – С. 3-8.

35. Найдыш В.М. Формование выпуклых параболических обводов [Текст] / В.М. Найдыш // Прикладная геометрия и инженерная графика. – К., 1983. – Вып. 37. – С. 35-37.

36. Найдыш В.М. Дискретне геометричне моделювання – новий напрямок розвитку прикладної геометрії кривих та поверхонь [Текст] / В.М. Найдыш // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2001. – Вип. 70. – С. 50-55.

37. Найдыш В.М. Дискретна сплайнова інтерполяція [Текст] / В.М. Найдыш // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2002. – Т. 15. – С. 3-8.

38. Найдыш В.М. Актуальные проблемы дискретного геометрического моделирования [Текст] / В.М. Найдыш // Геометричне та комп'ютерне моделювання. – Х.: ХДУХТ, 2005. – Вип.13. – С. 7-16

39. Найдыш В.М. Використання барицентричних координат при локальному згущенні ділянок дискретно поданої кривої [Текст] / В.М. Найдыш, Е.А. Гавриленко // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2003. – Т. 18.– С. 40-44.

40. Найдыш В.М. Зміст та особливості варіативного дискретного геометричного моделювання [Текст] / В.М. Найдыш, А.В. Найдыш // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТА, 2006. – Т. 32– С. 3-14.

41. Найдыш А.В. Дискретна інтерполяція перехідних ділянок ДПК на основі розв'язання різницевих схем [Текст] / А.В. Найдыш, Д.В. Спірінцев // Праці Таврійського державного агротехнологічного університету. – Вип. 4 : Прикладна геометрія та інженерна графіка. – Мелітополь: ТДАТУ, 2008. – Т. 37. – С. 3-8.

42. Овчарук І.В. Зміна кількості базових ліній політканини із збереженням гладкості поверхні, що проектується [Текст] / І.В. Овчарук // Прикладна геометрія та інженерна графіка. – К.: КНУБА, 2004. – Вип. 74. – С. 141-145.

43. Овчарук І.В. Конструювання гладких каналових поверхонь на основі політканинних перетворень [Текст] / І.В. Овчарук // Проблеми сучасного підручника. – Київ – Бердянськ, 2004. – Вип. 5. – Ч. 2. – С. 132-137.

44. Осипов В.А. Проектирование, расчет и исследование замкнутых плоских обводов типа “шпангоут” [Текст] / В.А. Осипов, Ж.Г. Иванова, А.П. Фаина // Кибернетика графики и прикладная геометрия поверхностей: сб. науч. трудов. – М.: МАИ, 1970. – С. 9-17.

45. Основи прикладної дискретної геометрії [Текст]: Навч. посіб. / Найдиш В.М., Верещага В.М., Найдиш А.В., Малкіна В.М. – Мелітополь, 2007. – 193 с.

46. Пугачев Е.В. Дискретная интерполяция пространственных кривых [Текст] / Е.В. Пугачев // Прикладна геометрія та інженерна графіка. – К.: КДТУБА, 1997. – Вип. 62. – С. 90-93.

47. Пугачов Є.В. Дискретна згладжуюча апроксимація дискретно представлених плоских ліній [Текст] / Є.В. Пугачов // Прикладна геометрія та інженерна графіка. – К.: КДТУБА, 1997. – Вип. 62. – С.184-186.

48. Пугачов Є.В. Дискретна згладжуюча апроксимація дискретно представлених просторових ліній [Текст] / Є.В. Пугачов // Прикладна геометрія та інженерна графіка. – К.: КДТУБА, 1998. – Вип. 63. – С. 33-36.

49. Пустюльга С.І. Вплив геометричних характеристик автомобільних доріг на безпеку руху транспорту [Текст] / С.І. Пустюльга, В.Р. Самостян // Сучасні проблеми геометричного моделювання. Наукові нотатки. : міжвуз. зб. – Луцьк: ЛДТУ, 2008. – Вип. 22, ч. 2. – С. 294-298.

50. Расчет и построение контуров самолета на плазе [Текст] / Андреев В.А., Зворыкин В.А., Коноворов Л.А. и др. ; под общ. ред. Ленёкова С.С. – М.: Оборонгиз, 1960. – 492 с.

51. Романова Д. П. Интеграция ГИС с экологическим мониторингом для контроля качества воды в бассейне Днепра / Ю. В. Сидоренко, Д. П. Романова// Тези

доповідей IV Всеукраїнської науково-практичної конференції «Наукова Україна: проблеми сучасності та перспективи майбутнього», 21-22 лютого 2017. – Харків: 2017. — С.63-65

52. Романова Д.П. Полікоординатне моделювання поверхні забруднення / Ю. В. Сидоренко, Д. П. Романова // Тези доповідей XV Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів «Сучасні проблеми наукового забезпечення енергетики», 25-28 квітня 2017. – Київ: НТУУ «КПІ» «Політехніка», 2017. — С.81

53. Романова Д. П. Моделювання об'єктів за допомогою методів прикладної геометрії / Ю. В. Сидоренко, Д. П. Романова// Збірник наукових праць IV науково-практичної дистанційної конференції молодих вчених і фахівців з розробки програмного забезпечення «Сучасні аспекти розробки програмного забезпечення», 30 квітня 2017 р. – Черкаси: видавець Чабаненко Ю.А., 2017. – С.103-107

54. Романова Д. П. Прогнозування динаміки економічних процесів за допомогою методів фрактальної геометрії / О.В. Кривда, Ю. В. Сидоренко, Д. П. Романова // Економічний вісник Національного технічного університету України “Київський політехнічний інститут”. [Текст]: Зб. наук. праць. – Вип. 14. - К:НТУУ «КПІ», 2017. – С. – Назва з екрана. – Доступ: <http://ev.fmm.kpi.ua/issue/view/6460>.

55. Романова Д.П. Розрахунок векторів переміщення базису при полікоординатних перетвореннях/ Ю. В. Сидоренко, Д. П. Романова // Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 24–27 квітня 2018 р. У 2 т. –К. : "КПІ ім. Ігоря Сікорського", 2018. –Т.2. – С.76

56. Сидоренко Ю.В. Обчислення коефіцієнта об'ємної теплоємності при моделюванні процесів плавлення сплавів [Текст] / Ю.В. Сидоренко, В. А. Третяк // ScienceRise. - 2015. - № 5/2 (10). - С. 60-64.

57. Сидоренко Ю.В. Система моделювання геометричних об'єктів за допомогою політочкових перетворень [Текст] / Сидоренко Ю.В. // Міжвідомчий науково-технічний збірник „Прикладна геометрія та інженерна графіка”.-Вип.92.- К.:КНУБА, 2016р.- С.118-125.

58. Сидоренко Ю.В. Параметрична інтерполяційна функція Гауса [Текст] / Сидоренко Ю.В.// Комп'ютерне моделювання в хімії, технологіях і системах сталого розвитку - КМХТ - 2014: Збірник наукових статей Четвертої міжнар. наук.- прак. конф.- Київ:НТУУ "КПІ" , 2014-С.67-73.

59. Сидоренко Ю.В. Конструювання геометричних об'єктів засобами політочкових перетворень /Бадаєв Ю.І., Ю.В. Сидоренко // Прикладна геометрія та інженерна графіка. - К.:КДТУБА, 2000-Вип.66-с.44-47.

60. Сидоренко Ю.В. Побудова гладких ліній за допомогою параметризованих функцій Гаусса/ Ю.В. Сидоренко // Прикладна геометрія та інженерна графіка — К.:КДТУБА, 2001, вип.69 — С.63-67.

61. Чорна Л.С. Моделювання поверхонь на основі полікоординатних В-сплайнів [Текст] / Л.С. Чорна // Системні технології : рег. міжвуз. зб. наук. праць. — Дніпропетровськ, 2006. — Вип. 3(44). — С. 173-178.

62. Чорна Л.С. Просторові полікоординатні векторно-параметричні криві [Текст] / Л.С. Чорна // Прикладна геометрія та інженерна графіка. — К.: КНУБА, 2006. — Вип. 76. — С. 192-193.

63. Чорна Л.С. Полікоординатні векторно-параметричні криві та поверхні в геометричному моделюванні [Текст]: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.01.01. "Прикладна геометрія, інженерна графіка" / Л.С. Чорна. — К., 2006. — 20 с.

64. Шипулин В.Д. Введение в использование ArcGIS / В.Д. Шипулин // Учебно-методическое пособие (для студентов дневной формы обучения спец. 6.07090 «Геоинформационные системы и технологии»). Сост. Шипулин В.Д. - Харьков: ХНАГХ, 2005. — 258 с.

65. Шпикуляк О.О. Імітаційне моделювання геометричних об'єктів / Ю.В.Сидоренко, О. О. Шпикуляк // Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 24–27 квітня 2018 р. У 2 т. –К. : "КПІ ім. Ігоря Сікорського", 2018. –Т.2. – С.94.

66. Blokker P.C. Spreading and evaporation of petroleum products on water. Proc. 4th Int. Harbour Congress, Verslagboek, Antwerp, The Netherlands, 1964, p. 911-919.

67. Blumberg A. F, G. L. Mellor, 1987, A description of a three-dimensional coastal ocean circulation model. AGU, Washington, DC, 4, 1.
68. Braunschweig, F (2001) – Generalização de um modelo de circulação costeira para albufeiras, MSc. Thesis, Instituto Superior Técnico, Technical University of Lisbon
69. Brovchenko I., Kuschan A., Maderich V., Shliakhtun M., Koshebutsky V., Zheleznyak M. Model for oil spill simulation in the Black Sea // Proc. of the 3rd Int. Conf. Oil Spills, Oil Pollution and Remediation. 16-18 Sept., 2003, Bogazici Univ., Istanbul, 2003, p. 101-112.
70. Cancino, L. and R. Neves (1999) - Hydrodynamic and sediment suspension modelling in estuarine systems. Part II: Application to the Western Scheldt and Gironde estuaries, *Journal of Marine Systems* 22, 117-131
71. Cecilia Ambjorn Seatrack Web forecasts and backtracking of oil spills – an efficient tool to find illegal spills using AIS. US/EU-Baltic International Symposium, Tallinn, May 27-29 2008. IEEE/OES, p. 16.
72. Coelho H.S., Neves R.J.J., White M., Leitao P.C., Santos A.J. A model for ocean circulation on the Iberian coast // *J. of Marine Systems*. 2002. vol. 32, p. 153-179
73. Fay J.A. Physical processes in the spread of oil on a water surface. Proc. of the Joint Conference on Prevention and Control of Oil Spills, p.463 ~ 467, American Petroleum Institute, Washington. D.C., 1971.
74. Fay J.A. The spread of oil slicks on a calm sea. In *Oil on the Sea*, edited by D.Hoult, p.53 64, Plenum, New York, 1969. A"
75. Neves, R., H. Coelho, P. Leitão, H. Martins, and A. Santos (1998) - A numerical investigation of the slope current along the western European margin. In: Burgano V., Karatzas G., Payatakas A., Brebbia C., Gray W. and Pinder G. (Ed.), *Computational Methods in Water Resources XII*, 2, 369-376, 1998.
76. Obukhova V.S. Development of Applied Geometriy of Algebraic Surfaces of Higher Orders [Text] / V.S. Obukhova // *Interdepartment Collection of Proceedings "Applied Geometry and Graphics"*/ Issue No 70 (Special issue)/ Kyiv: KNUBA, 2002. – P. 61-67.

77. Pina, P. M. N (2001) – An Integrated Approach to Study the Tagus Estuary Water Quality. Tese de Mestrado, Universidade Técnica de Lisboa, Instituto Superior Técnico

78. Romanova D. P Investigation of the poly-coordinate basis form for the correct mapping of the object [Text] / D. P. Romanova, I. V. Sydorenko// Science and Technology of the XXI Century : the XVIII All-Ukrainian Students R&D Conference Proceeding, (Kyiv, December 07, 2017) / National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. – Part V. – Kyiv, 2017. – 83-84 p

79. Romanova Daria. The application of the fractal approach in the modeling of economic and educational processes [Електронний ресурс] / Olena Kryvda, Iuliia Sydorenko, Daria Romanova // Zborník vzdelávanie a spoločnosť. - prešovskej univerzity v prešove (slovensko), 2017. – p. 294-298

80. Starodetko E.A. Parametrization of Implicitly Specified Curves by Means of Power Series [Text] / E.A. Starodetko // Interdepartment Collection of Proceedings “Applied Geometry and Graphics”/ Issue No 70 (Special issue) / Kyiv: KNUBA, 2002. – P. 71-78.

81. Taboada J.J., R. Prego, M. Ruiz-Villarreal, P. Montero, M. Gómez-Gesteira, A. Santos and V. Pérez-Villar (1998) - Evaluation of the seasonal variations in the residual patterns in the Ría de Vigo (NWSpain) by means of a 3D baroclinic model, Estuarine Coastal and Shelf Science 47, pp. 661-670

82. Tolman, H. L, 1999, User manual and system documentation of WAVEWATCH-III version 1.18. NOAA / NWS / NCEP / OMB Technical Note 166.

Villarreal, M.R., P. Montero, R. Prego, J.J. Taboada, P. Leitao, M. GómezGesteira, M. de Castro and V. Pérez-Villar (2000) - Water Circulation in the Ria de Pontevedra under estuarine conditions using a 3d hydrodynamical model, submitted to Est. Coast. and Shelf Sc.

ДОДАТОК А

Акт впровадження

Моделювання процесів деформації геометричних об'єктів

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТР32277_18М

Аркушів 1

2018

"Затверджую"

директор

(посада)

Роміна Т.М.

(п.і.б., підпис)

АКТ ВПРОВАДЖЕННЯ

результатів дипломної роботи магістра Шпикуляк О.О. на тему «Моделювання процесів деформації геометричних об'єктів», яка виконана в Національному технічному університеті України «Київський політехнічний інститут ім. Ігоря Сікорського» (НТУУ «КПІ ім. Ігоря Сікорського») 2018 р.

Нами, представниками кафедри автоматизації проектування енергетичних процесів і систем НТУУ «КПІ ім. Ігоря Сікорського» та ТОВ «ВІЗАВІ СОЛЮШНЗ », даний акт складено про те, що для використання в розробках спеціалізованого програмного забезпечення ТОВ «ВІЗАВІ СОЛЮШНЗ» прийняті результати дипломної роботи магістра Шпикуляк О.О., а саме програмне забезпечення – система консолідації даних у вигляді запису на CD, а також документацію програмного супроводу.

Представник кафедри АПЕПС НТУУ «КПІ ім. Ігоря Сікорського»

Керівник дипломної роботи

(підпис)

Сидоренко Ю.В.

Представник ТОВ «ВІЗАВІ СОЛЮШНЗ »

Директор

(підпис)

Роміна Т.М.